

webMethods EDIINT Module Installation and User's Guide

Version 8.0

December 2009

This document applies to webMethods EDIINT Module Version 8.0 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2000–2009 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

The name Software AG, webMethods, and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products." This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Table of Contents

About This Guide	7
Document Conventions	7
Documentation Installation	7
Online Information	8
1. Installing webMethods EDIINT Module	11
Overview	12
Requirements	12
Installing EDIINT Module 8.0 SP1	12
Upgrading EDIINT Module 6.5.2 to 8.0 SP1	13
Upgrading EDIINT Module 8.0 to 8.0 SP1	14
Uninstalling EDIINT Module 8.0 SP1	15
2. Concepts	17
What Is EDIINT?	18
webMethods EDIINT Module Introduction	18
webMethods EDIINT Module Support for EDIINT	19
EDIINT Interoperability Tests	19
How the EDIINT Module Relates to Other webMethods Components	20
Features Provided for EDIINT Processing	21
Inbound EDIINT Processing with AS1 and AS2	21
EDIINT Client	22
Processing Inbound EDIINT Documents	22
Processing Inbound EDIINT MDNs	25
Outbound EDIINT Processing with AS1 and AS2	26
Sending Outbound EDIINT Documents	27
Sending Outbound EDIINT MDNs	29
Run-Time Processing with EDIINT AS3	30
Run-Time Processing for Host Partners	31
Host Partner Sending an AS3 Message to a Client Partner	31
Host Partner Retrieving an AS3 Message or MDN from a Client Partner	31
Run-Time Processing for Client Partners	32
Client Partner Sending an AS3 Message to the Host Partner	32
Client Partner Retrieving an AS3 Message or MDN from the Host Partner	32
Using a Business Process to Send Outbound EDIINT Documents	33
3. Before You Can Transport Documents Using EDIINT	35
Overview	36
Including EDIINT Information in Profiles	36
External IDs Tab of the Profile	37
Behavior of External ID Matching	37

Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3	37
Using the EDIINT ID Match Option	38
Delivery Settings Tab of the Profile	39
Extended Fields Tab of the Profile	40
Certificates Tab of the Profile	42
Configuring SMTP Settings to Enable EDIINT Message Exchange	43
Configuring Inbound EDIINT SMTP Settings	43
Configuring Outbound EDIINT SMTP Settings	45
Configuring Your System to Support EDIINT AS3	45
Creating Directories for Uploading/Downloading	45
Creating Trading Partner Agreements (TPAs) for AS3 Support	46
Configuring Whether Trading Networks Is to Process Payloads	53
Configuring How to Handle Asynchronous MDNs	56
Specifying How Long to Wait before Sending an Asynchronous MDN	56
Processing Payloads when Asynchronous MDNs Are Not Sent	56
Trading Networks Objects Provided for EDIINT	57
TN Document Types	57
Document Attributes	58
Extended Fields	59
Processing Rules	59
Services Invoked by Processing Rules	60
4. Creating a Client to Submit a Document Using EDIINT	61
Introduction	62
Content Types to Use	62
Setting the Input Variables for the wm.EDIINT:send Service	62
5. Processing Inbound EDIINT Documents and MDNs	63
Processing Inbound EDIINT Documents	64
Before You Can Process Inbound EDIINT Documents	64
Example of an EDIINT Document Posted by HTTP	64
Creating Outbound MDNs that Acknowledge Receipt of an EDIINT Document	65
Setting Up to Process Payloads from EDIINT Documents	65
How the S/MIME Type Profile Field Affects Processing Payloads	66
Processing Inbound EDIINT MDNs	67
Before You Can Process Inbound EDIINT MDNs	67
Example of an EDIINT MDN Posted by HTTP	67
6. Using EDIINT to Deliver Outbound Documents	69
Before You Can Deliver Outbound EDIINT Documents	70
Setting the S/MIME Type of the Outbound EDIINT Document	70
Using the wm.EDIINT:send Service to Send EDIINT Documents	70
7. Viewing and Managing Information About EDIINT Documents and MDNs	73
Viewing Information about EDIINT Documents and MDNs	74

Viewing the User Status Associated with a Document	74
Viewing Related Documents	75
Resubmitting EDIINT Outbound Transactions	76
8. webMethods EDIINT Module Services	77
Overview	78
Summary of Elements in this Folder	78
wm.EDIINT:receive	78
wm.EDIINT:retrieveAS3Message	79
wm.EDIINT:send	80
A. Glossary	85

About This Guide

This guide is for users of the webMethods EDIINT Module. It provides an overview of the webMethods EDIINT Module and its features.

Document Conventions

Convention	Description
Bold	Identifies elements on a user interface.
Narrow font	Identifies storage locations for services on webMethods Integration Server, using the convention folder.subfolder.service.
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Documentation Installation


You can download the product documentation using the Software AG Installer. Depending on the release of the webMethods product suite, the location of the downloaded documentation will be as shown in the table below.

For webMethods...	The documentation is downloaded to...
6.x	The installation directory of each product.
7.x	A central directory named _documentation in the main installation directory (webMethods by default).

For webMethods...	The documentation is downloaded to...
8.x	A central directory named <code>_documentation</code> in the main installation directory (Software AG by default).

Online Information

You can find additional information about Software AG products at the locations listed below.

 Note: The Empower Product Support Web site and the Software AG Documentation Web site replace Software AG ServLine24 and webMethods Advantage.

If you want to...	Go to...
Access the latest version of product documentation.	Software AG Documentation Web site http://documentation.softwareag.com
Find information about product releases and tools that you can use to resolve problems. See the Knowledge Center to:	Empower Product Support Web site https://empower.softwareag.com
<ul style="list-style-type: none"> ■ Read technical articles and papers. ■ Download fixes and service packs. ■ Learn about critical alerts. 	
See the Products area to:	
<ul style="list-style-type: none"> ■ Download products. ■ Get information about product availability. ■ Access older versions of product documentation. ■ Submit feature/enhancement requests. 	

If you want to...	Go to...
■ Access additional articles, demos, and tutorials.	Software AG Developer Community for webMethods
■ Obtain technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with Software AG technology.	http://communities.softwareag.com/webmethods
■ Use the online discussion forums to exchange best practices and chat with other experts.	
■ Expand your knowledge about product documentation, code samples, articles, online seminars, and tutorials.	
■ Link to external Web sites that discuss open standards and many Web technology topics.	
■ See how other customers are streamlining their operations with technology from Software AG.	


1 Installing webMethods EDIINT Module

■ Overview	12
■ Requirements	12
■ Installing EDIINT Module 8.0 SP1	12
■ Upgrading EDIINT Module 6.5.2 to 8.0 SP1	13
■ Upgrading EDIINT Module 8.0 to 8.0 SP1	14
■ Uninstalling EDIINT Module 8.0 SP1	15

Overview

This chapter, in conjunction with the *Software AG Installation Guide*, explains how to install, upgrade, and uninstall webMethods EDIINT Module.

If you are installing webMethods EDIINT Module 8.0 at the same time you are installing other webMethods products, such as webMethods Integration Server, webMethods Trading Networks, and My webMethods Server, see the *Software AG Installation Guide* for instructions on installing those products.

 Important! webMethods EDIINT Module 8.0 requires webMethods Trading Networks to be installed. You must upgrade all your webMethods products at the same time; you cannot use 6.5.2 products with 8.0 products. For more information about upgrading your webMethods products, see the *webMethods Upgrade Guide*.


Software AG strongly recommends that you upgrade and migrate in a controlled test environment and test that installation for proper operation before upgrading and migrating your production environment.

Requirements

For a list of the operating systems and webMethods products that webMethods EDIINT Module 8.0 supports, see the *webMethods eStandards Module System Requirements*, available in the webMethods area of the [Software AG Documentation Web site](#).

Installing EDIINT Module 8.0 SP1

This section provides only instructions that are specific to installing EDIINT Module 8.0 SP1. For complete instructions on using the Software AG Installer, see the *Software AG Installation Guide*.

 Note: EDIINT Module stores data in Trading Networks tables. If you are installing EDIINT Module in a clustered environment, you must install EDIINT Module and Trading Networks on each Integration Server in the cluster, and each installation must be identical. For information on running Trading Networks in a clustered environment, see information about configuring Trading Networks in the *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.


► To install EDIINT Module 8.0 SP1

- 1 If you are going to install EDIINT Module 8.0 SP1 on an already installed Integration Server, shut down the Integration Server. Make sure that the version of Integration Server is 8.0.
- 2 Download the Software AG Installer from the Empower Product Support Web site at <https://empower.softwareag.com>.
- 3 Start the Software AG Installer wizard.
- 4 In the Release list, click Software AG Products 8.x. Provide your Software AG Empower user name and password. Click Next. The installer uses the user name and password to connect to the installer server and download the products for which you have purchased licenses.
- 5 Specify the installation directory as follows:
 - If you are installing EDIINT Module 8.0 SP1 on an existing Integration Server, specify the installation directory that contains the host Integration Server.
 - If you are installing both the host Integration Server and EDIINT Module 8.0 SP1, specify the installation directory to use. (The default is Software AG.)
- 6 In the product selection list, go to eStandards and select webMethods EDIINT Module 8.0 SP1. The installer installs the WmEDIINT package in the *IntegrationServer_directory\packages* directory.

You can select any required webMethods components you have not installed, such as Integration Server, Trading Networks, and EDI Module. For a list of required products and their versions, see the *webMethods eStandards Module System Requirements*, available in the webMethods area of the [Software AG Documentation Web site](#).


- 7 Start the Integration Server on which you installed EDIINT Module 8.0 SP1.

Upgrading EDIINT Module 6.5.2 to 8.0 SP1

 Important! webMethods EDIINT Module 8.0 requires webMethods Trading Networks to be installed. You must upgrade all your webMethods products at the same time; you cannot use 6.5.2 products with 8.0 products. For more information about upgrading your webMethods products, see the *webMethods Upgrade Guide*.

Software AG strongly recommends that you upgrade and migrate in a controlled test environment and test that installation for proper operation before upgrading and migrating your production environment.


▶ To upgrade EDIINT Module 6.5.2 to 8.0 SP1

- 1 Shut down all webMethods products and all other applications that are running on the machine on which you are going to install EDIINT Module 8.0 SP1.
- 2 If all products, applications, and business processes are not shut down, the installer will not be able to replace key files that are locked by the operating system.
- 3 Install EDIINT Module 8.0 SP1 as described in [“Installing EDIINT Module 8.0 SP1” on page 12](#). For the installation directory, specify a new installation directory.
- 4 Edit the `IntegrationServer_directory\packages\WmEDIINT\config\properties.cnf` file in your EDIINT Module 8.0 SP1 installation directory to copy the property values assigned in your EDIINT Module 6.5.2 installation.
- 5 Migrate the data and TPAs from webMethods Trading Networks. Follow the instructions in the *webMethods Upgrade Guide* to migrate data from Trading Networks.
 Important! When migrating from version 6.5 to 8.0 SP1, the value of S/MIME Type field is not migrated.
- 6 Run the service, `wm.EDIINT.doc:migrateSMimeType` in Integration Server Administrator to migrate the value of S/MIME Type field from version 6.5 to the SMIME Type field in 8.0 SP1.

Upgrading EDIINT Module 8.0 to 8.0 SP1


This upgrade procedure explains how to install EDIINT Module 8.0 SP1 over EDIINT Module 8.0.

▶ To upgrade EDIINT Module 8.0 to 8.0 SP1

- 1 Shut down all webMethods products and all other applications that are running on the machine on which you are going to install EDIINT Module 8.0 SP1. In addition, if any business processes are running, wait for them to complete normally or use Monitor to stop them.
 Important! If all products, applications, and business processes are not shut down, the installer will not be able to replace key files that are locked by the operating system.
- 2 Back up the EDIINT Module 8.0 installation directory:
`IntegrationServer_directory\packages\WmEDIINT`.
- 3 Install EDIINT Module 8.0 SP1. For the installation directory, specify the EDIINT Module 8.0 installation directory. For the installation procedure, see [“Installing EDIINT Module 8.0 SP1” on page 12](#).

- 4 Copy the *IntegrationServer_directory*\packages\WmEDIINT\config\properties.cnf file from your 8.0 back-up installation to the same directory in your 8.0 SP1 installation.
- 5 Migrate the data and TPAs from webMethods Trading Networks. Follow the instructions in the *webMethods Upgrade Guide* to migrate data from Trading Networks.

Uninstalling EDIINT Module 8.0 SP1

 Important! This section provides instructions that are specific to uninstalling webMethods EDIINT Module 8.0 SP1. For complete instructions on using the Software AG Uninstaller, see the *Software AG Installation Guide*.

 To uninstall EDIINT Module 8.0 SP1

- 1 Shut down the Integration Server that hosts EDIINT Module 8.0 SP1.
- 2 Start the Software AG Uninstaller, as follows:

System	Instructions
Windows	In the Add or Remove Programs window, select Software AG Products 8.x <i>installation_directory</i> , where <i>installation_directory</i> is the installation directory of the Integration Server on which EDIINT Module 8.0 SP1 is installed.
UNIX	Go to the <i>installation_directory</i> \bin directory of the installation that includes the Integration Server on which EDIINT Module 8.0 SP1 is installed and enter <code>uninstall (wizard)</code> or <code>uninstall -console</code> (console mode).

- 3 In the Select products to uninstall list, select eStandards > webMethods EDIINT Module 8.0 SP1.
- 4 The Software AG Uninstaller does not delete files that you have created or configuration files associated with EDIINT Module 8.0 SP1, nor does it delete the directory structure that contains those files. If you do not want to save those files, go to the *IntegrationServer_directory*\packages directory and delete the WmEDIINT package directories.
- 5 Restart the Integration Server from which you uninstalled webMethods EDIINT Module 8.0 SP1.

2 Concepts

■ What Is EDIINT?	18
■ webMethods EDIINT Module Introduction	18
■ webMethods EDIINT Module Support for EDIINT	19
■ EDIINT Interoperability Tests	19
■ How the EDIINT Module Relates to Other webMethods Components	20
■ Features Provided for EDIINT Processing	21
■ Inbound EDIINT Processing with AS1 and AS2	21
■ Outbound EDIINT Processing with AS1 and AS2	26
■ Run-Time Processing with EDIINT AS3	30
■ Using a Business Process to Send Outbound EDIINT Documents	33

What Is EDIINT?

EDIINT stands for “Electronic Data Interchange-Internet Integration,” or “EDI over the Internet.” EDIINT is a standard defined by the Internet Engineering Task Force (IETF) that defines a protocol for using the Internet to securely exchange business documents (EDI, XML, or other). The three versions of the EDIINT standard include:

- EDIINT AS1 (EDIINT Applicability Statement 1) that uses SMTP (e-mail) to transport documents
- EDIINT AS2 (EDIINT Applicability Statement 2) that uses HTTP (or HTTP/S) to transport documents
- EDIINT AS3 (EDIINT Applicability Statement 3) that uses FTPS (FTP over SSL) to transport documents

All versions support digital signatures, encryption, and signed receipts. The EDIINT standard specifies how to exchange business documents over the Internet in a secure, reliable, non-repudiable way. It does not specify how to validate or process the business documents that are transported.

webMethods EDIINT Module Introduction

The webMethods EDIINT Module (EDIINT Module) adds support for the EDIINT exchange protocol. Documents using the EDIINT protocol are processed through Trading Networks. As a result, if you want to use the EDIINT protocol, you must use Trading Networks.

The EDIINT Module supports EDIINT AS1 (SMTP), EDIINT AS2 (HTTP), and EDIINT AS3 (FTPS) messages, including MDN (receipt) exchange. The EDIINT Module exclusively uses the S/MIME version 2 cryptographic format to package, encrypt, and provide a digital signature to outbound data, as well as to unpack, decrypt, and verify the authenticity of inbound data.

You can use EDIINT to transport both EDI and non-EDI formatted (for example, XML or custom format) documents.

- When you use the EDIINT transport for EDI documents, you must also have the webMethods EDI Module installed. The EDIINT Module passes EDI documents to Trading Networks, which in turn allows the EDI Module to process the EDI document using the functions of the EDI Module packages (WmEDI and WmEDIforTN).
- When you use the EDIINT transport for non-EDI documents, the EDIINT Module passes the documents to Trading Networks, and Trading Networks processes them based on logic you define in Trading Networks.

webMethods EDIINT Module Support for EDIINT

The EDIINT Module supports the following EDIINT features:

- Securely exchanges business documents using EDIINT AS1, EDIINT AS2, and EDIINT AS3.
- Exclusively uses the S/MIME version 2 cryptographic format to package, compress, encrypt, and provide a digital signature to outbound data, as well as to unpack, decrypt, and verify the authenticity of inbound data.
 - 📄 Note: The S/MIME (Secure/Multipurpose Internet Mail Extensions) standard specifies formats and procedures for providing the cryptographic security services of message authentication, integrity, non-repudiation of origin, and confidentiality.
- Uses the SHA-1 hash algorithm to sign outbound messages, and verifies inbound messages that were signed with either SHA-1 or MD5.
- Enables you to set encryption types and key lengths for each of your trading partners using the extended fields in the partner's Trading Networks profile. The choices include: TripleDES, DES, RC2 (40 bits), RC2 (64 bits), and RC2 (128 bits).
- Provides the standard outbound encryption permutations (signed, encrypted, signed and encrypted, or plain) at the send-service level.
- Can send receipts of received messages back to the sender as well as receive message receipts. EDIINT receipts are known as MDNs (message disposition notifications). The EDIINT Module can send and receive synchronous or asynchronous, signed or unsigned MDNs.

EDIINT Interoperability Tests

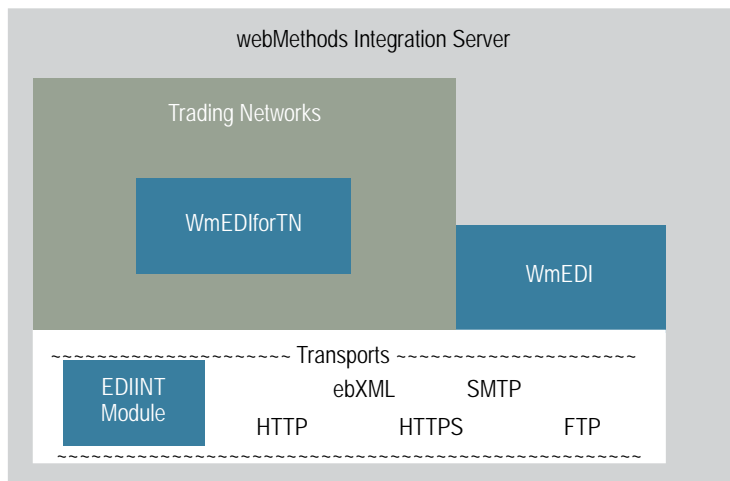
The webMethods EDIINT Module successfully completed the eBusinessReady interoperability testing for EDIINT AS1, EDIINT AS2, and EDIINT AS3. The eBusinessReady interoperability testing is an industry-neutral software testing program under joint partnership of the Uniform Code Council, Inc. (UCC) and Drummond Group, Inc. (DGI). For more information about the interoperability tests, refer to <http://www.eBusinessReady.org>.

Software AG has participated in all past EDIINT AS2 interoperability tests hosted by the Drummond Group.

How the EDIINT Module Relates to Other webMethods Components

When you install the webMethods EDIINT Module, the WmEDIINT package is installed into the Integration Server. The following diagram illustrates how the EDIINT Module fits into the webMethods product suite architecture.

EDIINT Module and the webMethods product suite architecture



- Integration Server is the underlying foundation.
- Trading Networks is a webMethods component that enables your enterprise to link with other companies (buyers, suppliers, strategic partners) and marketplaces to form a business-to-business trading network. Trading Networks is required to use the EDIINT Module. For more information about Trading Networks, see *Understanding webMethods B2B: webMethods Trading Networks Concepts Guide*, *Managing B2B Integrations: webMethods Trading Networks User's Guide*, and *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.
- EDIINT Module comprises the following two packages:
 - The WmEDI package is the basic functionality that provides support for the EDI standard.
 - The WmEDIforTN package allows for the interaction between the WmEDI package and Trading Networks. This interaction allows you to use Trading Networks as a gateway for EDI document exchange. The EDIINT Module uses the functionality of Trading Networks to provide additional features, such as support for VANs, reconciling FAs, and batching the sending of EDI documents.
- EDIINT Module contains the support for the EDIINT exchange protocol as described in [“webMethods EDIINT Module Introduction”](#) on page 18.

Features Provided for EDIINT Processing

The webMethods EDIINT Module provides the following to support EDIINT processing:

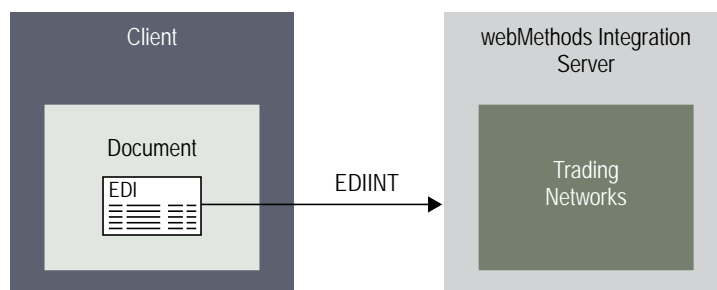
- Content handlers to recognize inbound EDIINT documents and MDNs
- Built-in services that you use to send outbound EDIINT documents and MDNs and receive inbound EDIINT documents and MDNs
- TN document types that are automatically installed into Trading Networks, allow Trading Networks to recognize inbound EDIINT documents and MDNs, and are set up so all EDIINT documents and MDNs are automatically saved to the Trading Networks database
- Extended profile fields that are automatically installed into Trading Networks and that are available in Trading Networks profiles for you to supply partner-specific information needed for EDIINT transport-level processing
- Processing rules that are automatically installed into Trading Networks and are used to perform the EDIINT transport-level processing
- Delivery services that are automatically registered with Trading Networks and used for delivering EDIINT documents and MDNs

For more information about how the EDIINT Module uses the items listed above for EDIINT inbound and outbound processing, see [“Inbound EDIINT Processing with AS1 and AS2” on page 21](#) and [“Outbound EDIINT Processing with AS1 and AS2” on page 26](#).

Inbound EDIINT Processing with AS1 and AS2

For inbound processing, a client sends a document to the Integration Server using the EDIINT exchange protocol. The document is processed in the Integration Server using services provided with the EDIINT Module and by using Trading Networks.

Inbound EDIINT processing



EDIINT Client

Use the EDIINT Module to create the client to send documents using EDIINT. If you are *not* using webMethods software on the client side, see documentation for the EDIINT software that you are using.

When the client sends the EDIINT document to the Integration Server, it must associate the inbound document with a content type that the EDIINT Module recognizes, for example, `multipart/signed`. When the Integration Server receives a document that has an EDIINT content type, it passes the document to the appropriate EDIINT content handler, which was installed when you installed the EDIINT Module.

The EDIINT content handler passes the document to the service the client specifies. To use the EDIINT exchange protocol, the client must specify the `wm.EDIINT:receive` service. The `wm.EDIINT:receive` service is a built-in service provided with the EDIINT Module. For more information about processing the EDIINT document, see [“Processing Inbound EDIINT Documents”](#) below.

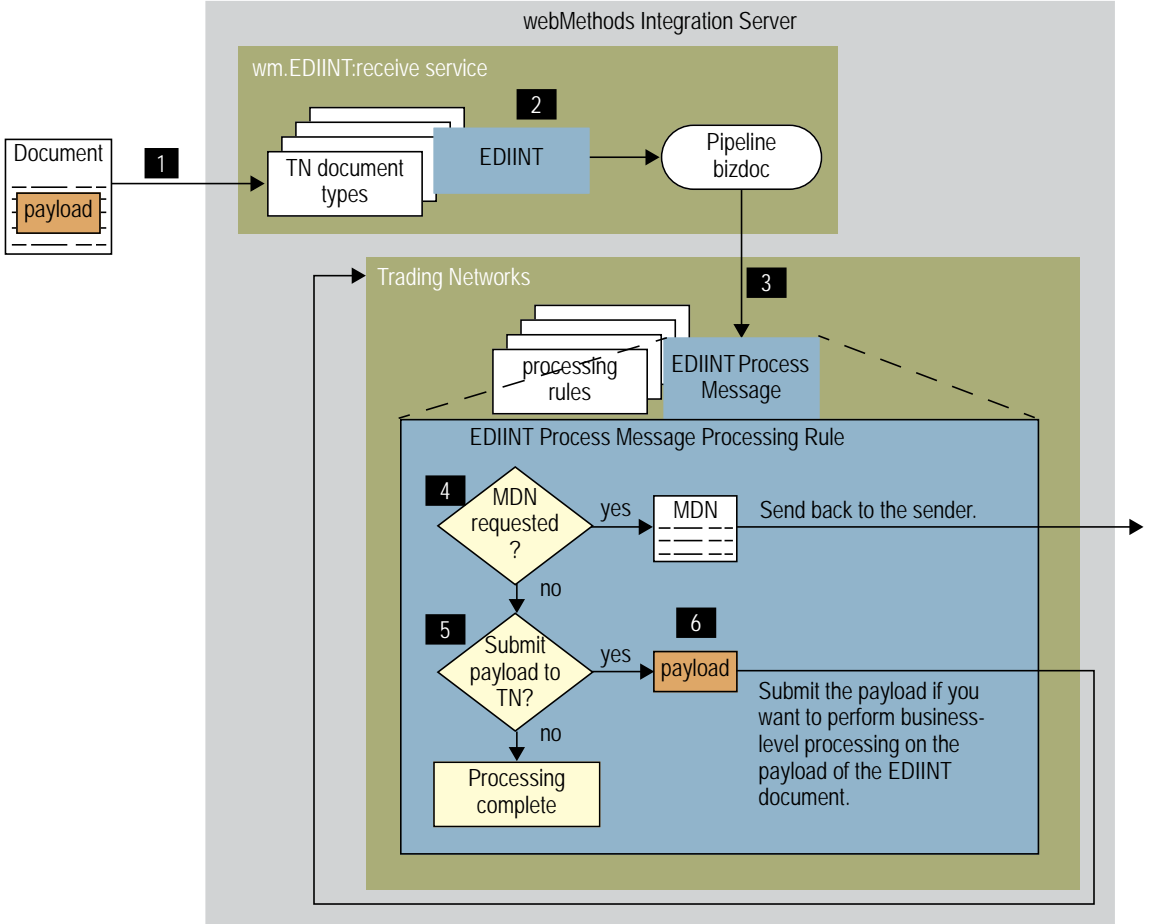
Processing Inbound EDIINT Documents

The EDIINT Module provides all logic required to perform the transport-level processing for inbound EDIINT documents. You can use Trading Networks to add business-level processing for the payloads of your EDIINT documents.

 Note: For information about how the EDIINT Module processes inbound MDNs, see [“Processing Inbound EDIINT MDNs”](#) on page 25.

The following diagram illustrates how the EDIINT Module performs transport-level processing for an inbound EDIINT document. The diagram also shows how, if you want to perform business-level processing, you can submit the payload of the EDIINT document to Trading Networks for further processing.

Processing an inbound EDIINT document



Step	Description
------	-------------

1 The wm.EDIINT:receive service accepts the inbound document.

2 The wm.EDIINT:receive service uses the TN document types to determine the type of document. The document matches the EDIINT TN document type that is installed into Trading Networks when you install the EDIINT Module.

After determining the TN document type, the wm.EDIINT:receive service forms a BizDocEnvelope for the inbound document and places it in the pipeline in the bizdoc variable. A BizDocEnvelope contains the original document and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to ProcessMsg.

After forming the BizDocEnvelope, the wm.EDIINT:receive service sends BizDocEnvelope to Trading Networks for processing.

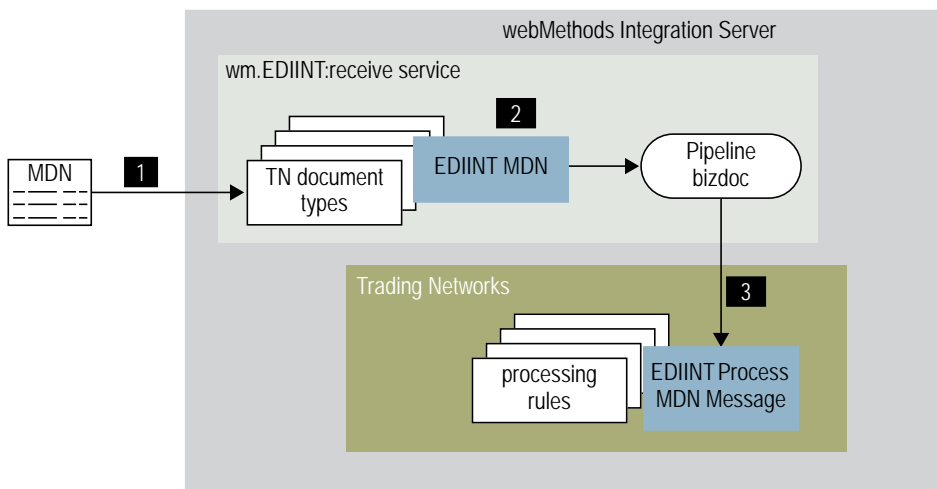
Step	Description
3	<p>Trading Networks determines the processing rule to use for the document. For inbound EDIINT documents, Trading Networks uses the EDIINT Process Message processing rule that is installed into Trading Networks when you install the EDIINT Module. It selects this processing rule because the TN document type is EDIINT and the user status is ProcessMsg.</p> <p>This processing rule performs the Execute a Service action to invoke the <code>wm.EDIINT.rules:processMsg</code> service. The service processes the message by opening the MIME or S/MIME package and then decrypting and/or verifying the signature of the message.</p>
The remaining steps represent actions specified in the EDIINT Process Message processing rule.	
4	<p>The <code>wm.EDIINT.rules:processMsg</code> service determines whether the sender of the EDIINT document requested an MDN.</p> <ul style="list-style-type: none">■ If the sender requested an MDN, the <code>wm.EDIINT.rules:processMsg</code> service determines whether the sender has requested a signed or an unsigned MDN. The service creates the appropriate type of MDN and sends the MDN back to Trading Networks for delivery. Trading Networks can deliver the MDN synchronously or asynchronously. For more information about how Trading Networks delivers the MDN, see “Sending Outbound EDIINT MDNs” on page 29. After sending the document to Trading Networks, continue with the next step.■ If the sender did not request an MDN, continue with the next step.
5	<p>The <code>wm.EDIINT.rules:processMsg</code> service invokes the <code>wm.EDIINT.rules:processPayload</code> service, which determines whether you want to send the payload of the EDIINT document to Trading Networks for processing.</p> <p>You define whether you want the EDIINT Module to send EDIINT payloads to Trading Networks for processing when you configure the EDIINT Module.</p>

Step	Description
6	<p>If the payload is to be sent to Trading Networks for processing (for example, to perform business-level logic), submit the payload. The payload can be either an EDI document or a non-EDI document (for example, an XML document).</p> <ul style="list-style-type: none"> EDI documents. If the payload is an EDI document, the EDI Module processes the document. When the payload is an EDI document, you must set up the EDI Module and Trading Networks to process the EDI document. For example, use the EDI Module to install TN document types for the EDI documents and create a processing rule to process the EDI document. For more information about how the EDI Module processes EDI documents, see how to use the EDI Module with Trading Networks in the <i>webMethods EDI Module Concepts Guide</i>. Non-EDI documents. If the payload is not an EDI document, you must define the TN document types that Trading Networks can use to recognize the payload and the processing rule that Trading Networks should use to process the document.

Processing Inbound EDIINT MDNs

The following diagram illustrates how the EDIINT Module processes inbound MDNs.

Processing inbound MDNs



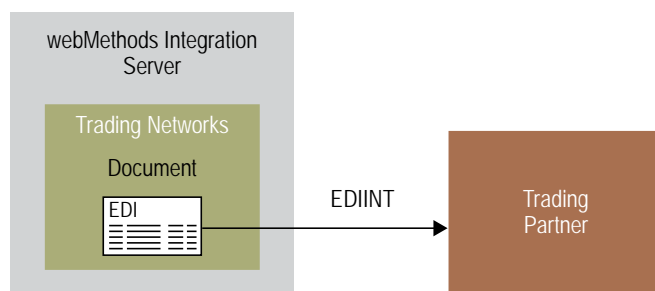
Step	Description
1	The <code>wm.EDIINT:receive</code> service accepts the inbound MDN.
2	<p>The <code>wm.EDIINT:receive</code> service uses the TN document types to determine the type of document. The MDN matches the EDIINT MDN TN document type that is installed into Trading Networks when you install the EDIINT Module.</p> <p>After determining the TN document type, the <code>wm.EDIINT:receive</code> service forms a <code>BizDocEnvelope</code> for the inbound MDN and places it in the pipeline in the <code>bizdoc</code> variable. A <code>BizDocEnvelope</code> contains the MDN and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to <code>ProcessMDNMsg</code>.</p> <p>After forming the <code>BizDocEnvelope</code>, the <code>wm.EDIINT:receive</code> service sends <code>BizDocEnvelope</code> to Trading Networks for processing.</p>
3	<p>Trading Networks determines the processing rule to use for the MDN. For inbound MDNs, Trading Networks uses the EDIINT Process MDN Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT MDN and the user status is <code>ProcessMDNMsg</code>.</p> <p>This processing rule performs the <code>Execute a Service</code> action to invoke the <code>wm.EDIINT.rules:processMDN</code> service, which processes the EDIINT MDN.</p>

Outbound EDIINT Processing with AS1 and AS2

The documents you want to send using EDIINT can be EDI documents or non-EDI documents. The EDIINT standard specifies requirements for how to “package” a document for transport and how to transport the document.

To package the document and transport it, you use services provided with the EDIINT Module along with Trading Networks.

Outbound EDIINT processing



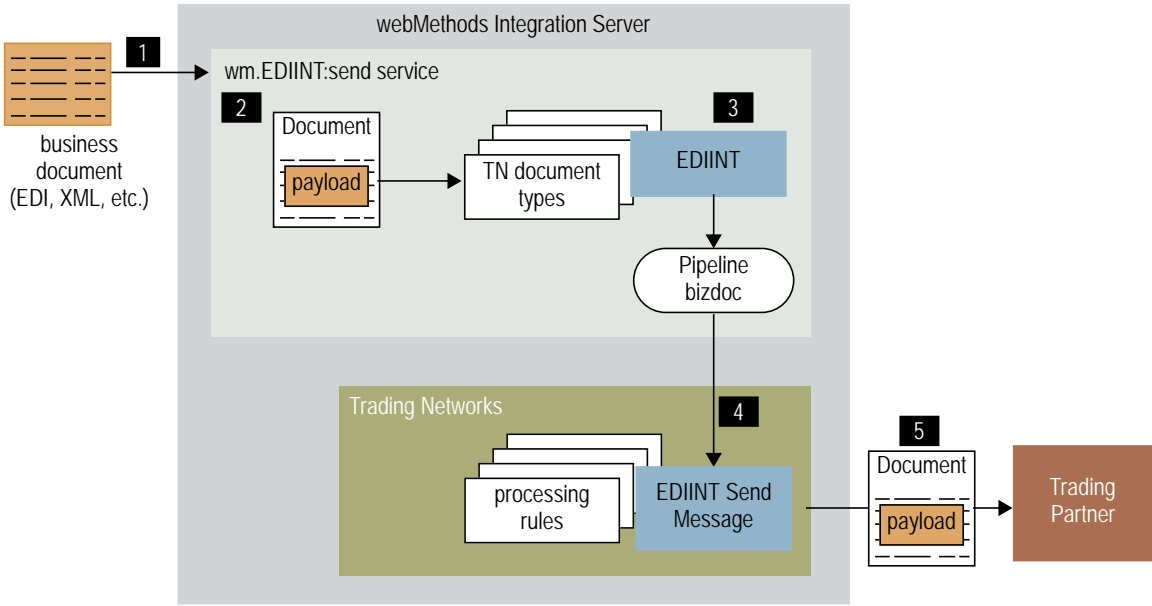
Sending Outbound EDIINT Documents

The EDIINT Module provides all logic required to perform the transport-level processing for sending outbound EDIINT documents.

Note: For information about how to send outbound MDNs, see “Sending Outbound EDIINT MDNs” on page 29.

The following diagram illustrates how to use the EDIINT Module to send an outbound EDIINT document.

Sending outbound EDIINT documents



Step	Description
1	You invoke the <code>wm.EDIINT:send</code> service to send a document to the EDIINT Module to be packaged as an EDIINT document and delivered to the receiving trading partner. The inputs of <code>wm.EDIINT:send</code> include the level of encryption to use, whether to compress the data, whether you are requesting an MDN and if so what kind, and how to deliver the document.
2	Using the input information that you provide, the <code>wm.EDIINT:send</code> service creates the EDIINT document (that is, an EDIINT MIME or S/MIME message).

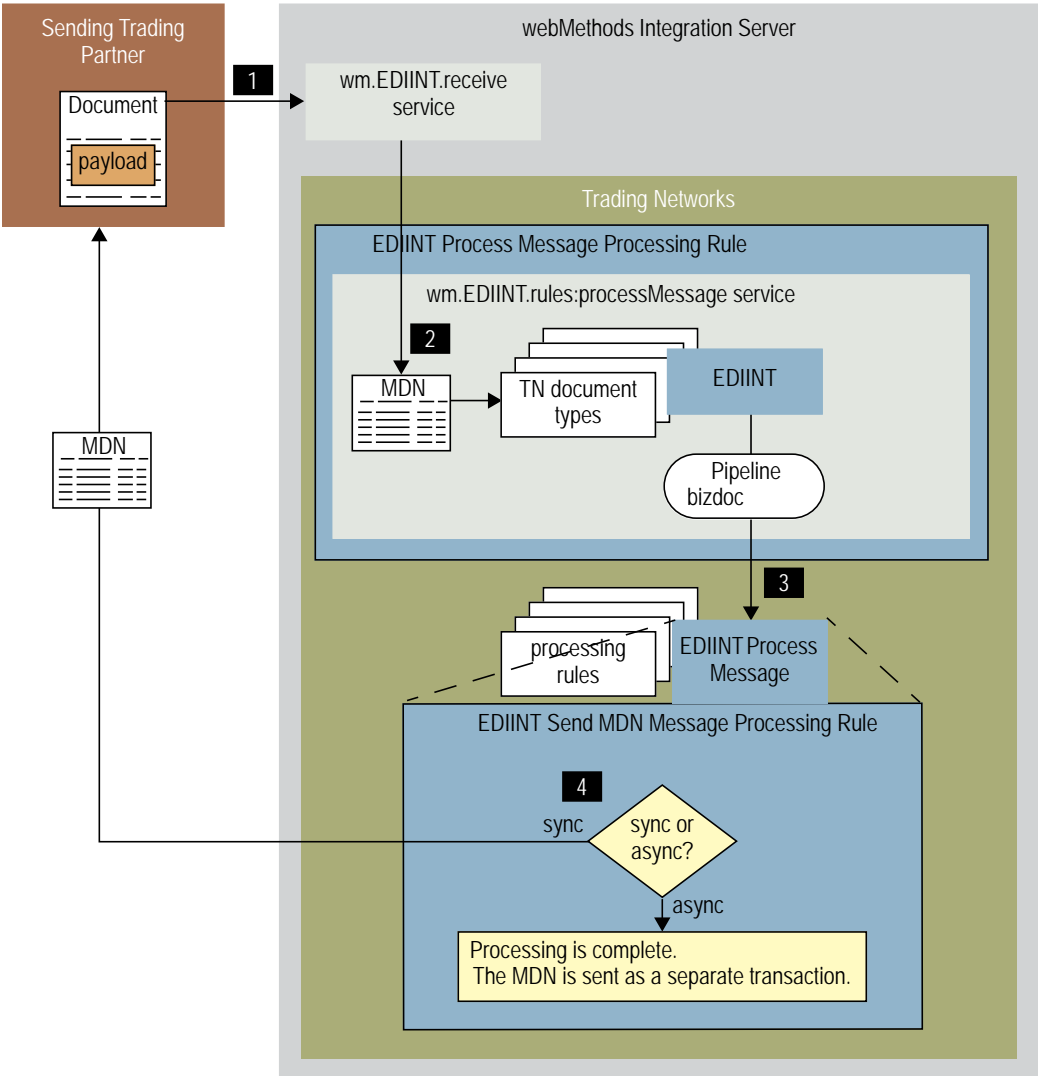
Step	Description
3	<p>The <code>wm.EDIINT:send</code> service uses the TN document types to determine the type of document. The document matches the EDIINT TN document type that is installed into Trading Networks when you install the EDIINT Module.</p> <p>After determining the TN document type, <code>wm.EDIINT:send</code> forms a <code>BizDocEnvelope</code> for the inbound document and places it in the pipeline in the <code>bizdoc</code> variable. A <code>BizDocEnvelope</code> contains the original document and includes additional information that Trading Networks requires for routing and processing the document. One piece of information that Trading Networks can use in the selection of a processing rule is the user status. The EDIINT recognizer sets the user status to <code>SendMsg</code>.</p> <p>After forming the <code>BizDocEnvelope</code>, the <code>wm.EDIINT:send</code> service sends <code>BizDocEnvelope</code> to Trading Networks for processing.</p>
4	<p>Trading Networks determines the processing rule to use for the document. For outbound EDIINT documents, Trading Networks uses the EDIINT Send Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT and the user status is <code>SendMsg</code>.</p> <p>This processing rule performs the <code>Execute a Service</code> action to invoke the <code>wm.EDIINT.rules:sendMsg</code> service.</p>
5	<p>The <code>wm.EDIINT.rules:sendMsg</code> service delivers the document by invoking the <code>wm.EDIINT.delivery:deliveryDocument</code> service, which in turn delivers the document to the appropriate trading partner.</p>

Sending Outbound EDIINT MDNs

The EDIINT Module automatically sends an outbound MDN when it receives an inbound EDIINT document that requests an MDN. Based on how the sender of the inbound EDIINT document requests the MDN to be sent, the EDIINT Module can send the MDN either synchronously or asynchronously. When the EDIINT Module sends an MDN synchronously, it sends the MDN using the same HTTP connection as that of the inbound EDIINT document. Otherwise, it sends the MDN as a separate transaction.

The following diagram illustrates how the EDIINT Module sends an outbound MDN.

Sending an outbound MDN



Step	Description
1	<p>The sender sends an EDIINT document to the <code>wm.EDIINT:receive</code> service with a request for an MDN.</p>
2	<p>The <code>wm.EDIINT:receive</code> service accepts the inbound EDIINT document and passes the document to Trading Networks. Trading Networks processes the document using the EDIINT Process Message processing rule. For details about inbound processing, see “Processing Inbound EDIINT Documents” on page 22.</p> <p>The EDIINT Process Message processing rule invokes the <code>wm.EDIINT.rules:processMsg</code> service to process the inbound EDIINT document. Because an MDN is requested, the <code>wm.EDIINT.rules:processMsg</code> service creates the MDN and performs document recognition on the MDN using the Trading Networks TN document types. The MDN matches the EDIINT MDN TN document type. After determining the TN document type, Trading Networks forms a BizDocEnvelope for the MDN and places it in the pipeline in the <code>bizdoc</code> variable. The Trading Networks user status for the MDN is set to <code>SendMDNMsg</code>. The <code>wm.EDIINT.rules:processMsg</code> service then passes the BizDocEnvelope into standard Trading Networks processing.</p>
3	<p>Trading Networks determines the processing rule to use for the MDN. For inbound MDNs, Trading Networks uses the EDIINT Send MDN Message processing rule that is installed into Trading Networks when you install the EDIINT Module. Trading Networks selects this processing rule because the TN document type is EDIINT MDN and the user status is <code>SendMDNMsg</code>.</p> <p>The EDIINT Send MDN Message processing rule performs the <code>Execute a Service</code> action to invoke the <code>wm.EDIINT.rules:sendMDN</code> service.</p>
4	<p>The <code>wm.EDIINT.rules:sendMDN</code> service determines what type of MDN the sender has requested (synchronous or asynchronous) and then sends the MDN accordingly:</p> <ul style="list-style-type: none">■ If the sender requested a synchronous MDN, the service returns a synchronous MDN to the sender using the same HTTP connection.■ If the sender requested an asynchronous MDN, the service invokes the <code>wm.EDIINT.delivery:deliveryDocument</code> service to send an asynchronous MDN as a separate transaction.

Run-Time Processing with EDIINT AS3

To exchange AS3 messages with a trading partner, you use an FTP server that is located either on your system or on your trading partner’s system. You use just one FTP server. The partner with the FTP server is referred to as the *host partner*.

The partner who accesses the host partner’s FTP server is referred to as the *client partner*. To retrieve the AS3 messages or files, the client partner needs to log in remotely as an FTP client.

To enable the partners to exchange AS3 messages, the host partner must provide the client partner with a particular set of specifications known as a choreography. The AS3 term *choreography* refers to the actions that occur between a client and an FTP server, and the FTP commands that enable those actions to occur. The choreography describes the means for delivering, retrieving, and deleting AS3 messages. It includes information about how an upload is communicated to the server as finished and available for a trading partner to download, such as renaming the file extension. In addition, it states whether the partner who downloads the message must send a delete command to clean up the file, or whether the message is removed through other means within the server.

The three categories of actions are:

- Server logon actions (secure or un-secure)

The EDIINT Module uses the secure FTP support provided by the Integration Server. Using this support, clients connect to remote FTP servers using Secure Sockets Layer (SSL).

- Document upload and download actions
- MDN upload and download actions

Each partner must specify the choreography information in a Trading Partner Agreement (TPA), as described in [“Creating Trading Partner Agreements \(TPAs\) for AS3 Support” on page 46](#).

Run-Time Processing for Host Partners

If you are a host partner (the partner with the FTP server on your Integration Server), you will send and retrieve AS3 messages as follows:

Host Partner Sending an AS3 Message to a Client Partner

- 1 The host partner invokes the `wm.EDIINT:send` service, which creates an AS3 message and uploads it to the `userFtpRoot\userhome\AS3\inbox` directory on the FTP server.
- 2 The client partner logs in to the host partner’s FTP server and retrieves the message from the host partner’s `userFtpRoot\userhome\AS3\inbox` directory. The client partner can optionally delete the message.

Host Partner Retrieving an AS3 Message or MDN from a Client Partner

- 1 The client partner logs in to the host partner’s FTP server and uploads the AS3 message or MDN to the host partner’s `userFtpRoot\userhome\AS3\outbox` directory.
- 2 EDIINT submits the AS3 message or MDN to Trading Networks to be processed.
- 3 EDIINT places an MDN response message (if the TPA specifies it) in the host partner’s `userFtpRoot\userhome\AS3\inbox` directory.

Run-Time Processing for Client Partners

If you are a client partner (the partner accessing a remote FTP server on an Integration Server), you will send and retrieve AS3 messages as follows:

Client Partner Sending an AS3 Message to the Host Partner

- 1 The client partner invokes the `wm.EDIINT:send` service, which creates an AS3 message and uploads it to the host partner's FTP server.
- 2 The host partner processes the AS3 message and *puts* an MDN (if the TPA specifies it) on the host partner's FTP server.

Client Partner Retrieving an AS3 Message or MDN from the Host Partner

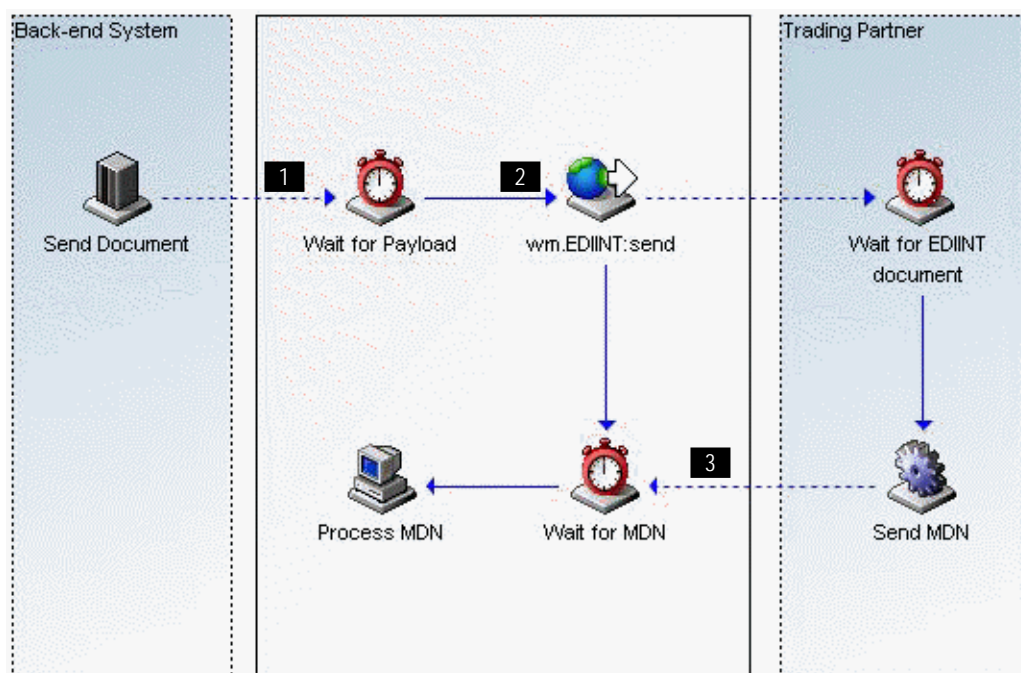
- 1 The client partner invokes the `wm.EDIINT:retrieveAS3Message` service, which logs in to the host partner's FTP server and downloads the AS3 message or MDN. EDIINT deletes the message if the TPA specifies it.
- 2 EDIINT submits the AS3 message or MDN to Trading Networks to be processed.
- 3 If the TPA requires an MDN response message, EDIINT logs in to the FTP server and uploads the MDN to the host partner's FTP server.

For information about configuring your system to support EDIINT AS3, see [“Configuring Your System to Support EDIINT AS3”](#) on page 45.

Using a Business Process to Send Outbound EDIINT Documents

You can design a process model that waits for a document that you want to send using the EDIINT transport. You can assign a conversation ID to the outbound EDIINT document. When its corresponding MDN is returned, the EDIINT Module assigns the MDN the same conversation ID, so the MDN can rejoin the conversation.

Sample process model that waits, sends an EDIINT document, and receives the corresponding MDN



Step	Description
1	The business process waits for a document, for example, from a back-end system. This is the document that you want to send using EDIINT.
2	The next step in the business process forms an EDIINT document with the back-end system document as the payload, and sends the EDIINT document to the trading partner. The step invokes the <code>wm.EDIINT:send</code> service to package the back-end system document as the payload of an EDIINT document. The <i>ConversationID</i> input parameter to the <code>wm.EDIINT:send</code> service is set to define the value to use for the conversation ID. It should be the same conversation ID that the back-end system document used.
3	The trading partner responds with an MDN. The EDIINT Module sets the conversation ID of the MDN to the value specified for the <i>ConversationID</i> input parameter in the preceding step. As a result, the MDN rejoins the correct business process.

3 Before You Can Transport Documents Using EDIINT

- Overview 36
- Including EDIINT Information in Profiles 36
- Configuring SMTP Settings to Enable EDIINT Message Exchange 43
- Configuring Your System to Support EDIINT AS3 45
- Configuring Whether Trading Networks Is to Process Payloads 53
- Configuring How to Handle Asynchronous MDNs 56
- Trading Networks Objects Provided for EDIINT 57

Overview

Before you can transport EDI or non-EDI documents using the EDIINT transport, perform the following tasks:

- Ensure that the partner profiles for senders and receivers contain the information that the EDIINT Module requires, as described in [“Including EDIINT Information in Profiles”](#) on page 36.
- If you want to use EDIINT AS1 (SMTP), or if you want to use EDIINT AS2 and have partners send MDNs via SMTP, configure the SMTP settings to enable the EDIINT Module to send and receive EDIINT documents via SMTP, as described in [“Configuring SMTP Settings to Enable EDIINT Message Exchange”](#) on page 43.
- If you want to use EDIINT AS3 (FTPS), configure your system as described in [“Configuring Your System to Support EDIINT AS3”](#) on page 45.
 - 📄 Note: If you use HTTP, HTTPS, or FTPS, see *Administering webMethods Integration Server* or the *webMethods Integration Server Administrator’s Online Help* for instructions about how to add a port. No EDIINT-specific settings are required to add an HTTP, HTTPS, or FTPS port.
- If you want the EDIINT Module to send the payload of the document to Trading Networks for further processing, configure the EDIINT Module appropriately, as described in [“Configuring Whether Trading Networks Is to Process Payloads”](#) on page 53.
- If you want to control the way in which the EDIINT Module handles asynchronous MDNs, configure the EDIINT Module as described in [“Configuring How to Handle Asynchronous MDNs”](#) on page 56.

Including EDIINT Information in Profiles

To use the EDIINT transport when exchanging documents between partners (senders and receivers), the partners’ profiles in Trading Networks must contain EDIINT information that the EDIINT Module requires.

For steps to create or modify profiles, see the chapter about partner profiles in *Building B2B Integrations: webMethods Trading Networks Administrator’s Guide*.

The following sections identify the specific information that you supply to use the EDIINT transport. The sections do *not* describe all profile fields. See *Building B2B Integrations: webMethods Trading Networks Administrator’s Guide* for descriptions of profile fields that are *not* listed in the sections below.

External IDs Tab of the Profile


Provide values for the following fields on the External IDs tab of the Partner Profile page in My webMethods:

Corporate Tab field	Description
ID Type and Value	<p>Select all versions of the EDIINT standard that the partner's corporation uses: EDIINT AS1, EDIINT AS2, and/or EDIINT AS3. You may also select any number of other external ID types, such as User Defined.</p> <p>Then, for each selected external ID type, specify a value to identify the partner. You may assign the same external ID value for multiple external ID types within the same profile. For example, if the partner uses EDIINT AS2 and EDIINT AS3, you can specify the same identification value for both types.</p> <p>The external ID type and its value correspond to the values in the EDIINT document headers "From" and "To" (for AS1), "AS2-From" and "AS2-To" (for AS2), and "AS3-From" and "AS3-To" (for AS3).</p>

Behavior of External ID Matching

If you used a version of the EDIINT Module *prior* to 6.5.2 and you want to use your existing profiles in 8.0, you have two options:

- Update your existing profiles so they use the external ID types EDIINT AS1, EDIINT AS2, or EDIINT AS3 instead of the existing external ID types. For details, see ["Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3"](#) on page 37.
- OR-
- If you want to continue using the external ID types of your existing profiles without having to change your existing profiles, turn off the new EDIINT ID Match option to make the EDIINT Module behave as it did prior to version 8.0. For details, see ["Using the EDIINT ID Match Option"](#) on page 38.

 Important! You *must* either update your existing profiles or turn off the EDIINT ID Match option. Otherwise, you may get unpredictable results when trying to match external IDs.


Updating Your Existing Profiles to Use EDIINT AS1, EDIINT AS2, and EDIINT AS3

Prior to version 6.5.2, the EDIINT Module only looked at the external ID value to find a matching profile when sending or receiving documents; it ignored the external ID *type*. Thus, each external ID value was required to be unique; a value could not be duplicated within a single profile or in any other profile.

With version 8.0, the EDIINT Module looks at both the external ID value *and* the external ID type in order to find a matching profile. This means you may assign the same external ID value for multiple external ID types within the same profile or in any other profile (or the values may be unique).

By default, when you send an EDIINT document the `wm.EDIINT:send` service tries to match the document's "To" header (for example, `AS2-To: 987654321`) with the external ID type and value defined in a partner profile (for example, EDIINT AS2 and 987654321).

Similarly, when you receive an EDIINT document the `wm.EDIINT:receive` service tries to match the value of its sender ID input parameter (which specifies both type and value) with a partner profile's external ID type and value.

 Important! If you use the EDIINT AS1, EDIINT AS2, or EDIINT AS3 types in your profiles, make sure the EDIINT ID Match option is selected (this is the default). If you turn off this option, you may get unpredictable results when trying to match external IDs. For details, see ["Using the EDIINT ID Match Option" on page 38](#).

Using the EDIINT ID Match Option

If you do *not* want to use the external ID types EDIINT AS1, EDIINT AS2, or EDIINT AS3 in your profiles, turn *off* the EDIINT ID Match option. Otherwise, leave this option turned on.

If you fail to set this option appropriately, you may get unpredictable results when trying to match external IDs. For example, if you define a profile with three external ID types that have identical values as follows:

External ID type	Value
EDIINT AS2	987654321
EDIINT AS3	987654321
User Defined	987654321

and you send an EDIINT document with the following headers:

AS2-From: 123456789

AS2-To: 987654321

the EDIINT ID Match option controls the external ID matching as follows:

When EDIINT ID Match is...	The matching external ID...
On	Is EDIINT AS2.
Off	Could be any of the three IDs; you cannot predict which one will match.

▶ To set the EDIINT ID Match option

1 Open the Integration Server Administrator if it is not already open.


2 In the Solutions menu of the navigation panel, click EDIINT.

The Integration Server Administrator opens a new browser window to display the EDIINT Module home page.

3 In the navigation panel of the EDIINT Module home page, click Configuration.

4 Specify how to match external IDs as follows:

Do this...	To have the EDI Module try to match...
Select the EDIINT ID Match check box	Both the external ID value and the external ID type.
Clear the EDIINT ID Match check box	Only the external ID value.

 Note: You can also set this option in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, set *wm.EDIINT.EDIINTIDMatch* to true or false.

Delivery Settings Tab of the Profile

On the Delivery Settings tab of the Partner Profile page in My webMethods, specify the delivery methods that EDIINT requires. The *wm.EDIINT:send* service uses the information that you specify on this tab to send outbound EDIINT messages and MDNs.

If you are using...	Do this...
EDIINT AS1	Specify at least one of the following delivery methods: 1) Primary E-mail, 2) Secondary E-mail.
	Note: You must define the delivery method in both the sender and receiver profiles. The e-mail address in the sender's profile is used for the "From" address and the e-mail address in the receiver's profile is used for the delivery e-mail address ("To").
EDIINT AS2	Specify at least one of the following delivery methods: 1) Primary HTTP, 2) Secondary HTTP, 3) Primary HTTPS, 4) Secondary HTTPS.

 Note: EDIINT AS3 is an available delivery method as well. For details, see ["Configuring Your System to Support EDIINT AS3" on page 45.](#)

Extended Fields Tab of the Profile

The following table lists the Extended Fields tab fields you should supply for EDIINT on the Partner Profile page in My webMethods.

Extended Fields Tab Field	Description
AS1MDNURL	<p>If you are using EDIINT AS1, specify the e-mail address that is to accept inbound AS1 EDIINT MDNs (for example, <code>receiver@company.com</code>).</p> <p>An inbound AS1 MDN is by definition asynchronous because it is not returned using the same connection as that of the originally sent document.</p>
AS2MDNURL	<p>If you are using EDIINT AS2, specify the URL that is to accept inbound AS2 EDIINT MDNs.</p> <ul style="list-style-type: none"> ■ To accept MDNs via HTTP, specify a URL that includes the <code>/invoke/</code> element to invoke the <code>wm.EDIINT:receive</code> service. For example (where <i>host:port</i> would be an actual host and port number): <pre>http://host:port/invoke/wm.EDIINT/receive</pre> ■ To accept AS2 MDNs via SMTP, specify a URI similar to the following: <pre>mailto:receiver@company.com</pre> <p>An inbound AS2 MDN could be synchronous (HTTP only) or asynchronous (SMTP or HTTP). A synchronous MDN is returned using the same HTTP connection as that of the originally sent document.</p>
Encryption Algorithm	<p>Specify the encryption option to use for outbound EDIINT messages.</p> <p>The choices are TripleDES (the default), DES, RC2 40 (40 bits), RC2 64 (64 bits), and RC2 128 (128 bits).</p>

Extended Fields Tab Field	Description
S/MIME Type	<p>The S/MIME type to use for payloads sent and received by the trading partner. Specify one of the following:</p> <ul style="list-style-type: none"> ■ plain, which means payloads are neither signed nor encrypted. ■ signed, which means payloads are signed. ■ encrypted, which means payloads are encrypted. ■ signedAndEncrypted, which means payloads are signed and encrypted. This is the default. <p>For more information about how the S/MIME Type extended profile field is used during inbound processing, see “How the S/MIME Type Profile Field Affects Processing Payloads” on page 66. For more information about how the S/MIME Type extended profile field is used during outbound processing, see “Setting the S/MIME Type of the Outbound EDIINT Document” on page 70.</p>
Compression	<p>Specify <code>true</code> to compress the outbound message before signing and/or encrypting it, or <code>false</code> (the default) to sign and/or encrypt the message without compressing it.</p>
Delivery Method	<p>Specify the delivery method you want to use to send the EDIINT document.</p> <ul style="list-style-type: none"> ■ For EDIINT AS1, specify one of the following: <ul style="list-style-type: none"> ■ PrimarySMTP (corresponds to the Trading Networks Primary E-mail delivery method) ■ SecondarySMTP (corresponds to the Trading Networks Secondary E-mail delivery method) ■ For EDIINT AS2, specify one of the following: <ul style="list-style-type: none"> ■ PrimaryHTTP ■ SecondaryHTTP ■ PrimaryHTTPS ■ SecondaryHTTPS ■ For EDIINT AS3, specify AS3.
Request MDN	<p>Specify whether you want the receiver to return an MDN. Specify one of the following:</p> <ul style="list-style-type: none"> ■ none, to <i>not</i> request a return MDN. ■ synchronousMDN, to request a return synchronous MDN. ■ asynchronousMDN, to request a return asynchronous MDN.

Extended Fields Tab Field	Description
Request Signed Receipt	Specify <code>true</code> to request a signed MDN or <code>false</code> to request a plain (unsigned) MDN.
FTPUserName	If you are using EDIINT AS3 and you are the hosting partner (that is, your partner will access your FTP server), specify your partner's user name so that EDIINT can place AS3 messages and MDNs in the <code>userFtpRoot\username\AS3\inbox</code> directory. Your partner will download the messages and MDNs from this directory. For more information about the inbox directory, see "Creating Directories for Uploading/Downloading" on page 45.

Certificates Tab of the Profile

The following table lists the information that you should supply for EDIINT on the Certificates tab of the Partner Profile page in My webMethods.


Subtabs on the Security Tab	Description
Sign/Verify	<p>If you want the EDIINT Module to be able to digitally sign outbound EDIINT documents, specify the certificates for your corporation along with your private key on the Sign/Verify tab of the profile.</p> <ul style="list-style-type: none">■ You can set up a default signing certificate information in the Enterprise profile.■ If you need to use a specific certificate to sign outbound documents for a particular partner, specify the certificate information on the Sign/Verify tab of that partner's profile. <p>If you expect to receive from a partner an EDIINT document with a digital signature, and you want the EDIINT Module to be able to verify the digital signature, specify verify certificates on the Sign/Verify tab in the partner's profile. You specify the certificates for your partner's corporation.</p>

Subtabs on the Security Tab	Description
Decrypt/Encrypt	<p>If you expect to receive encrypted EDIINT documents from partners, specify the certificates for your corporation along with your private key that you need to use to decrypt the documents on the Decrypt/Encrypt tab of the profile.</p> <ul style="list-style-type: none"> ■ You can set up default decrypting certificate information in the Enterprise profile. ■ If you need to use a specific certificate to decrypt inbound documents from a particular partner, specify the certificate information on the Decrypt/Encrypt tab of that partner's profile. <p>If you want the EDIINT Module to be able to encrypt outbound EDIINT documents for a partner, specify encrypt certificates on the Decrypt/Encrypt tab in the partner's profile. You specify the certificates for your partner's corporation.</p>

Configuring SMTP Settings to Enable EDIINT Message Exchange

To allow the EDIINT Module to receive and send documents using SMTP, you must configure inbound (for receiving) and outbound (for sending) SMTP settings. You must configure these settings if:

- You want to use EDIINT AS1.
- You want to use EDIINT AS2 and have partners send MDNs via SMTP.

 Important! Check your mailbox settings for message size limitations that could adversely affect your ability to receive or send large EDIINT documents.

Configuring Inbound EDIINT SMTP Settings


To receive an inbound EDIINT message or MDN via SMTP, use these general guidelines.

 To configure your system to be able to receive inbound EDIINT documents via SMTP

- 1 Set up an e-mail account with an e-mail service provider that supports either the POP3 or IMAP protocols.
- 2 From the Integration Server Administrator, add a webMethods/Email port that corresponds to the e-mail host that you established in the previous step. For instructions about how to add a port, see either *Administering webMethods Integration Server* or the *webMethods Integration Server Administrator's Online Help*.

The following table lists information you should specify when adding the port:

<u>In this section of the screen...</u>	<u>For this field...</u>	<u>Specify...</u>
Package	Package Name	Either <code>WmEDIINT</code> or <code>WmRoot</code> . <ul style="list-style-type: none">■ Specify <code>EDIINT</code> if you want the port disabled when the <code>WmEDIINT</code> package is disabled.■ Specify <code>WmRoot</code> if you want the port available whenever the server is running.
Server Information	all fields	Information about the e-mail host that you established in the previous step.
Security	Require authorization within message	No
	Run services as user	A user account with administrator authority, e.g., Administrator.
Message Processing	Global Service	<code>wm.EDIINT:receive</code>
	Default service	leave blank
	Invoke service for each part of multipart message	No
	Include email headers when passing message to content handler	Yes

 Note: The EDIINT Module does not use the fields in the Message Processing section of the screen that are not listed in the table above. Leave the fields set to their default values.


- 3 Enable the port.
- 4 Edit the port's Access Mode and click Set Access Mode to Allow by Default.

Configuring Outbound EDIINT SMTP Settings

To send outbound EDIINT documents or MDNs via SMTP, you must configure the name of the SMTP server you want to use for outbound EDIINT documents. You only need to perform this procedure if you want to use EDIINT AS1.

▶ To enable outbound EDIINT SMTP transport

- 1 Open the Integration Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDIINT. The Integration Server Administrator opens a new browser window to display the EDIINT Module home page.
- 3 In the navigation panel of the EDIINT Module home page, click Configuration.
- 4 In the SMTP Server field, type the name of your SMTP server.
- 5 Click Save Changes.

 Note: You can also set this option in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, set *wm.EDIINT.SMTPHost* to the name of your SMTP server.

Configuring Your System to Support EDIINT AS3

Configuring your system to support EDIINT AS3 involves creating directories for uploading and downloading, and then creating Trading Partner Agreements (TPAs) for AS3 support.

Creating Directories for Uploading/Downloading

The host partner must create the following directories in the Integration Server, under the default FTP root directory that the Integration Server creates at startup:

`userFtpRoot\userhome\AS3`

where *userhome* is the user's FTP home directory.

 Note: To rename the default userFtpRoot directory, use the Integration Server configuration parameter `watt.server.userFtpRoot`. For details, see *Administering webMethods Integration Server*.

The host partner must then create the following directories under the AS3 directory:

- `userFtpRoot\userhome\AS3\inbox` — The directory from which the host partner will download AS3 messages.
- `userFtpRoot\userhome\AS3\outbox` — The directory to which the host partner will upload AS3 messages.

Creating Trading Partner Agreements (TPAs) for AS3 Support

A Trading Partner Agreement (TPA) in Trading Networks is a set of parameters that you can use to govern how documents are exchanged between two trading partners. One partner fulfills the sender role during document exchange, and the other partner fulfills the receiver role. Both the sender and receiver in a TPA must be a partner who has an existing profile in your Trading Networks system.

Each TPA must specify a unique combination of the following:

- A partner that represents the originator of a send or retrieve operation
- The partner of the originator
- The Agreement ID (the type of the TPA); to support AS3 message exchange, use the predefined Agreement ID EDIINTAS3

You might have multiple TPAs for a pair of trading partners. For example, if PartnerA is the originator of a send or retrieve operation, you would define the following TPA:

TPA Field	Value
Sender	PartnerA
Receiver	PartnerB
Agreement ID (type of TPA)	EDIINTAS3


Conversely, if PartnerB is the originator of a send or retrieve operation, you would define the following TPA:



TPA Field	Value
Sender	PartnerB
Receiver	PartnerA
Agreement ID (type of TPA)	EDIINTAS3



You define TPAs on the Agreement Details screen of the Trading Networks Console, as described below.

To define a TPA

- 1 Start the Trading Networks Console. If you need procedures for this step, see *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.
- 2 Select **View > Agreements**.
- 3 Perform *one* of the following procedures on the Agreements screen to create a new TPA.

To create this type of TPA...	Do this...
New—The TPA fields are empty.	Select Agreements > New. Trading Networks displays the Agreement Details screen to allow you to fill in the information for the TPA.
Duplicate—The TPA fields are filled with values from the TPA agreements that you duplicate. You can update any or all fields from the duplicated TPA agreement.	<ol style="list-style-type: none"> 1 Click the row containing the name of the TPA agreement that you want to duplicate. 2 Select Agreements > Duplicate. Trading Networks displays the Agreement Details screen.
You can use this function to create a template TPA.	
3 Fill in the following fields on the Agreement Details screen for the TPA you want to create:	
 Note: The Sender, Receiver, and Agreement ID fields must be unique for each TPA. The values of these three fields together uniquely identify a TPA. After you create a TPA, you <i>cannot</i> change or update these fields of the TPA.	

For this TPA field...	Specify...
Sender	<p>Name of the trading partner that has the originator role in the transaction that the TPA will govern. Type in the name of the partner or click the Select () button to select the partner from the Partner Selection Dialog. This list includes your own profile (Enterprise).</p> <p>For EDI, to create a template that you will duplicate to create other TPAs, you can use the default value of Unknown.</p>
Receiver	<p>Name of the trading partner that receives the send or retrieve operations from Sender. Type in the name of the partner or click the Select () button to select the partner from the Partner Selection Dialog. This list includes your own profile (Enterprise).</p> <p>For EDI, to create a template that you will duplicate to create other TPAs, you can use the default value of Unknown.</p>

For this TPA field...	Specify...
Agreement ID	<p>Specify the Agreement ID <code>EDIINTAS3</code>, which indicates that the type of agreement between the two partners is an AS3 agreement using EDIINT.</p> <hr/> <p>Note: You will not be able to continue creating a TPA unless you specify the Agreement ID.</p> <hr/>
IS Document Type	<p>Specify the IS document type <code>wm.EDIINT.TPA:EDIINTAS3</code>. This IS document type defines the application-specific TPA data. The TPA data is used to govern the exchange of documents between the two partners.</p> <p>Alternatively, click Find IS Document Type () to browse the IS document types and select it. Trading Networks displays the data tree input values of the selected IS document type in the bottom panel of the Agreement Details screen. The right side of this panel displays the input values (TPA data inputs) to variables of the IS document type.</p>
Control Number	Accept the default value zero (0) or leave blank; EDIINT does not use this field.
Data Status	Whether you want to be able to modify the values of the TPA data of the IS document type. The data status is only applicable when the agreement status is Agreed.
Export Service	Leave blank; EDIINT does not use this field.
Initialization Service	<p>Specify the service <code>wm.EDIINT.TPA:initService</code>. This service populates the inputs to the variables in the IS document type <code>wm.EDIINT.TPA:EDIINTAS3</code> with default values.</p> <p>Type the name of the initialization service located on the server or click Find Service () to browse the services and select the one you want to use.</p>
Description	Optionally, type a description for the TPA. The description can be up to 1024 characters in length. There are no other restrictions.

For more information about creating TPAs, see *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.

- 4 Click the Set Inputs icon and provide values for the following upload input parameters for the document type `wm.EDIINT.TPA:EDIINTAS3`:

Upload Parameter	Description
AS3FTPServerLocation	<p>Indicates whether to upload/download AS3 messages to a remote or local FTP server.</p> <ul style="list-style-type: none"> ■ remote — Upload/download AS3 messages to/from the remote FTP server in the <code>FTPUpload</code> and <code>FTPDownload</code> fields. ■ local — Upload/download AS3 messages to/from the local FTP server. No other TPA fields will be used.
uploadService	Always select the <code>wm.EDIINT.delivery.defaultFTPUpload</code> service, which uploads AS3 files.
serverhost	Name or IP address of the FTP server.
serverport	Port number on which the FTP server listens for requests. Default: 21.
dataport	Optional. Listener port number of the data transfer channel, for example, 3345.
username	FTP user on the remote FTP server.
password	Password for the user specified in username.
account	Optional. The user name for an account on the FTP server. Specify a value if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user and password specified in username and password.
transfertype	<p>Specifies the type of the FTP data transfer mode:</p> <ul style="list-style-type: none"> ■ active — Default. Active FTP data transfer mode. ■ passive — Passive FTP data transfer mode.
encoding	Optional. Default character set for encoding data transferred during this session. Specify an IANA-registered character set, for example, <code>ISO-8859-1</code> . If you do not set encoding, the default JVM encoding is used.
timeout	Optional. Time (measured in seconds) to wait for a response from the FTP server before timing out and terminating the request. The default is to wait forever.
secureFTP	Type of the remote FTP server to connect to.
securedata	<p>Specifies whether to protect the FTP data channel.</p> <p><code>true</code> — Protect the FTP data channel.</p> <p><code>false</code> — Do not protect the FTP data channel.</p>

<u>Upload Parameter</u>	<u>Description</u>
auth	Authentication/security mechanism: <ul style="list-style-type: none">■ SSL■ TLS■ TLS-P
dirpath	Optional. The directory path to which AS3 messages are uploaded. If you do not specify a directory path, the current directory will be used.
fileExtension	Optional. The file extension to be assigned to the uploaded AS3 message file, for example, <code>msg</code> . AS3 message file names are generated using the following naming convention: <code>MMddhhmmSSss</code> where <code>MM</code> is month, <code>dd</code> is day, <code>hh</code> is hour, <code>mm</code> is minutes, and <code>SSss</code> is seconds. For example, a generated file name with the extension <code>msg</code> might be <code>122012002222.msg</code> .
renameTo	Optional. After uploading the AS3 message file, move it to a specified directory, and optionally rename their file extension. For example, if your FTP <code>put</code> command places the AS3 file in the <code>tmp</code> directory, and you want to move it to the <code>outbox</code> directory after uploading it, specify this command in the <code>renameTo</code> field: <code>/outbox/*</code> The wildcard character <code>*</code> is a placeholder for the file name. If you specified a file extension in the <code>fileExtension</code> field (for example, <code>msg</code>), you would specify this command in the <code>renameTo</code> field: <code>/outbox/*.msg</code> Optionally, you can rename the file extension in this field as well.
MDNDirpath	Optional. The directory to which MDNs are uploaded.

Upload Parameter	Description
MDNFileExtension	<p>Optional. The file extension to be assigned to the uploaded MDN file, for example, <code>mdn</code>.</p> <p>MDN file names are generated using the following naming convention:</p> <p><code>MMddhhmmSSss</code></p> <p>where <code>MM</code> is month, <code>dd</code> is day, <code>hh</code> is hour, <code>mm</code> is minutes, and <code>SSss</code> is seconds. For example, a generated file name with the extension <code>mdn</code> might be <code>122012002222.mdn</code>.</p>
MDNRenameTo	<p>Optional. After uploading the MDN files, move it to a specified directory, and optionally rename its file extension.</p> <p>For example, if your FTP <code>put</code> command places the MDN file in the <code>tmp</code> directory, and you want to move it to the <code>outbox</code> directory after uploading it, specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*</code></p> <p>The wildcard character <code>*</code> is a placeholder for the file name.</p> <p>If you specified a file extension in the <code>fileExtension</code> field (for example, <code>mdn</code>), you would specify this command in the <code>renameTo</code> field:</p> <p><code>/outbox/*.mdn</code></p> <p>Optionally, you can rename the file extension in this field as well.</p>
storeUnique	Optional. EDIINT assigns a unique file name after the file is uploaded.
transfermode	<p>Specifies the type of the FTP file transfer mode:</p> <ul style="list-style-type: none"> ■ binary (required for AS3) ■ ascii

- 5 Scroll down and provide values for the following download input parameters for `wm.EDIINT.TPA:EDIINTAS3`:

Download Parameter	Description
downloadService	Always select the <code>wm.EDIINT.delivery.defaultFTPDownload</code> service, which downloads AS3 files.
serverhost	Name or IP address of the FTP server.
serverport	Port number of the FTP server. Default: 21.

<u>Download Parameter</u>	<u>Description</u>
dataport	Optional. Listener port number of the data transfer channel, for example, 3345.
username	FTP user on the remote FTP server.
password	Password for the user specified in username.
account	Optional. The user name for an account on the FTP server. Specify a value if your FTP host requires account information. The account is defined in the FTP protocol to further identify the user and password specified in username and password.
transfertype	Specifies the type of the FTP data transfer mode: <ul style="list-style-type: none">■ active — Default. Active FTP data transfer mode.■ passive — Passive FTP data transfer mode.
encoding	Optional. Default character set for encoding data transferred during this session. Specify an IANA-registered character set, for example, ISO-8859-1. If you do not set encoding, the default JVM encoding is used.
timeout	Optional. Time (measured in seconds) to wait for a response from the FTP server before timing out and terminating the request. The default is to wait forever.
secureFTP	Type of the remote FTP server to connect to.
securedata	Specifies whether to protect the FTP data channel. true — Protect the FTP data channel. false — Do not protect the FTP data channel.
auth	Authentication/security mechanism: <ul style="list-style-type: none">■ SSL■ TLS■ TLS-P
dirpath	Optional. The directory path to which AS3 messages are downloaded. If you do not specify a directory path, the current directory will be used.
filenamepattern	Optional. AS3 message file pattern, for example, *.msg.
MDNDirpath	Optional. The directory to which MDNs are downloaded. You can specify either the path relative to dirpath or the absolute path.
MDNFilenamepattern	Optional. MDN message file pattern, for example, *.mdn.

<u>Download Parameter</u>	<u>Description</u>
deleteFile	Indicates whether the file is to be deleted after downloading it.
transfermode	Specifies the type of the FTP file transfer mode: <ul style="list-style-type: none">■ binary (required for AS3)■ ascii

6 Click OK to create the TPA.

Configuring Whether Trading Networks Is to Process Payloads

In addition to having the EDIINT Module perform transport-level processing for an entire inbound EDIINT document, you can perform further processing on the payload of the document by having the EDIINT Module send the payload to Trading Networks for separate processing. As installed, the EDIINT Module is configured to submit the payload to Trading Networks.

When the EDIINT Module is configured to send the payload to Trading Networks, the EDIINT Module submits the payload after it completes transport-level processing.

Use the EDI Module to process payloads that have one of the following content types:

- application/edi-X12
- application/EDIFACT
- application/XML
- application/edi-consent (which you can use to submit TRADACOMS payloads)


If you want to have the EDIINT Module submit payloads that have a different content type, you must provide your own service to process the payload and submit it to Trading Networks.

► To enable or disable Trading Networks payload processing

- 1 Open the Integration Server Administrator if it is not already open.
- 2 In the Solutions menu of the navigation panel, click EDIINT. The Integration Server Administrator opens a new browser window to display the EDIINT Module home page.
- 3 In the navigation panel of the EDIINT Module home page, click Configuration.

4 Enable or disable the EDIINT Module to or from submitting payloads to Trading Networks by doing one of the following:

- To enable, select the Submit payload to TN check box.
- To disable, clear the Submit payload to TN check box.

 Note: You can also set this option in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, set *wm.EDIINT.submitPayload* to true or false.

5 In the User Process Payload Service field, specify a service that you created to process payloads and submit them to Trading Networks. You *only* need to specify a service if:


- You selected the Submit payload to TN check box.

-AND-

- The content types of the inbound payloads are *not* application/edi-X12, application/EDIFACT, application/XML, or application/edi-consent.

The service you specify in the User Process Payload Service field must accept the following input variables:


<u>Input Variable</u>	<u>Description</u>
<i>stream</i>	InputStream The payload.
<i>contentType</i>	String The content type of the payload.
<i>EDIINTbizdoc</i>	Document The bizdoc that contains the original EDIINT message. For the structure of EDIINTbizdoc, see the <i>wm.tn.rec: BizDocEnvelope</i> service in the <i>webMethods Trading Networks Built-in Services Reference</i> .

 Note: You can also specify this service in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, specify the service in the *wm.EDIINT.userProcessPayloadService* variable.

The EDIINT Module ignores your settings in the following situations:

Conditions	Behavior of EDIINT Module
<ul style="list-style-type: none"> ■ You select the Submit payload to TN check box. ■ You specify a service in the User Process Payload Service field. 	<p>The EDI Module ignores the service you specify in the User Process Payload Service field.</p>
-AND-	<p>When you select the Submit payload to TN check box, the EDIINT Module <i>always</i> uses the EDI Module to process payloads</p>
<ul style="list-style-type: none"> ■ The inbound payload has a content type that <i>is</i> one of: <ul style="list-style-type: none"> ■ application/edi-X12 ■ application/EDIFACT ■ application/XML ■ application/edi-consent 	<p>that have one of the following content types:</p> <ul style="list-style-type: none"> ■ application/edi-X12 ■ application/EDIFACT ■ application/XML ■ application/edi-consent
<ul style="list-style-type: none"> ■ You select the Submit payload to TN check box. 	<p>The EDIINT Module ignores the check in the Submit payload to TN check box.</p>
<ul style="list-style-type: none"> ■ You do <i>not</i> specify a service in the User Process Payload Service field. 	<p>The EDIINT Module <i>cannot</i> submit a payload with an unsupported content type to Trading Networks. You must provide a service that submits payloads with unsupported content types.</p>
-AND-	
<ul style="list-style-type: none"> ■ The content type of the inbound payload is <i>not</i> one of: <ul style="list-style-type: none"> ■ application/edi-X12 ■ application/EDIFACT ■ application/XML ■ application/edi-consent 	
<ul style="list-style-type: none"> ■ You clear the Submit payload to TN check box. 	<p>The EDIINT Module ignores the service you specify in the User Process Payload Service field.</p>
-AND-	
<ul style="list-style-type: none"> ■ You specify a service in the User Process Payload Service field. 	<p>The EDIINT Module only invokes a service you specify to submit payloads to Trading Networks when you select the Submit payload to TN check box.</p>

6 Click Save Changes.

 Note: For information about how to set up the EDI Module and Trading Networks to process EDI documents, see the *webMethods EDI Module Installation and User's Guide*.

Configuring How to Handle Asynchronous MDNs

You can configure how the EDIINT Module should handle the sending of asynchronous MDNs in two ways:

- You can specify how long the EDIINT Module waits before sending an asynchronous MDN.
- You can specify that the EDIINT Module should process a payload even when the asynchronous MDN is not successfully sent.

Specifying How Long to Wait before Sending an Asynchronous MDN


By default, the EDIINT Module waits 30 seconds before sending an asynchronous MDN. However, this delay is not always sufficient. For example, some systems require the original HTTP connection to close before accepting an HTTP connection for the asynchronous MDN, and that process sometimes takes longer than 30 seconds.

▶ To increase the wait time before sending an asynchronous MDN


- 1 Open the EDIINT Module properties.cnf file in a text editor. This file is located in the following folder:

IntegrationServer_directory\packages\WmEDIINT\config\

- 2 Locate the `wm.EDIINT.waitInSeconds` property.
- 3 Set the value of this property to the number of seconds you want the EDIINT Module to wait before sending the asynchronous MDN, up to 60 seconds.

 Note: If you specify a value greater than 60 seconds, the EDIINT Module uses 60 seconds.

- 4 Save the properties.cnf file.
- 5 Reload the WmEDIINT package in Integration Server. For instructions, see *Administering webMethods Integration Server*.

 Note: You can also set this property in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, set `wm.EDIINT.waitInSeconds` to the number of seconds you want the EDIINT Module to wait before sending the asynchronous MDN.

Processing Payloads when Asynchronous MDNs Are Not Sent


By default, the EDIINT Module requires that an asynchronous MDN be sent successfully before it sends the payload message to Trading Networks for processing. As a result, if the asynchronous MDN cannot be successfully sent, the payload does not get processed.

► To specify payload processing even if an asynchronous MDN is not successfully sent

- 1 Open the EDIINT Module properties.cnf file in a text editor. This file is located in the following folder:

IntegrationServer_directory\packages\WmEDIINT\config\

- 2 Locate the `wm.EDIINT.processPayloadIfMDNNotSent` property.
- 3 Set the value of this property to `true`.
- 4 Save the properties.cnf file.
- 5 Reload the WmEDIINT package in Integration Server. For instructions, see *Administering webMethods Integration Server*.

 Note: You can also set this property in My webMethods: Administration > Integration > B2B Settings > Configure Properties. In the EDIINT Module Configuration Properties section, set `wm.EDIINT.processPayloadIfMDNNotSent` to `true` or `false`.

Trading Networks Objects Provided for EDIINT

When you install the EDIINT Module, Trading Networks objects (that is, TN document types, document attributes, extended profile fields, and processing rules) are installed in Trading Networks for you. This section describes the Trading Networks objects provided with the EDIINT Module.

The information in this section about the Trading Networks objects is for reference *only*. You should *not* alter the definitions of any of the Trading Networks objects.

TN Document Types

The following table describes the TN document types provided for EDIINT processing and the document attributes associated with each. For more information about the document attributes, see “[Document Attributes](#)” below.

TN document type name	Description	Associated document attributes
EDIINT	Trading Networks matches all EDIINT documents to this TN document type.	<ul style="list-style-type: none"> ■ EDIINT Message Type ■ EDIINT Message ID ■ EDIINT Message Digest ■ EDIINT Delivery URL

TN document type name	Description	Associated document attributes
EDIINT MDN	Trading Networks matches all EDIINT MDNs to this TN document type.	<ul style="list-style-type: none"> ■ EDIINT Message Type ■ EDIINT Message ID ■ EDIINT Delivery URL ■ EDIINT MDN Original Message ID ■ EDIINT MDN Received MIC ■ EDIINT MDN Disposition

Document Attributes

The following table describes the document attributes provided for EDIINT processing. The TN document types described in “[TN Document Types](#)” above extract these attributes from the EDIINT documents and MDNs.

This attribute name...	Represents...	And is extracted from...
EDIINT Message Type	The protocol that the EDIINT document uses: AS1, AS2, or AS3.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs
EDIINT Message ID	The value of the EDIINT Message-ID header, which is also used for the value of the Trading Networks Document ID system attribute.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs
EDIINT Message Digest	The message digest calculated for the EDIINT document.	<ul style="list-style-type: none"> ■ EDIINT documents
EDIINT Delivery URL	The destination URL or IP address from the EDIINT document.	<ul style="list-style-type: none"> ■ EDIINT documents ■ EDIINT MDNs
EDIINT MDN Original Message ID	The value of the EDIINT Message-ID header from the original EDIINT document for which the MDN is a receipt.	<ul style="list-style-type: none"> ■ EDIINT MDNs
EDIINT MDN Received MIC	The message digest calculated for the EDIINT MDN.	<ul style="list-style-type: none"> ■ EDIINT MDNs
EDIINT MDN Disposition	The results from processing the original EDIINT document for which the MDN is a receipt.	<ul style="list-style-type: none"> ■ EDIINT MDNs

Extended Fields

When you install the EDIINT Module, the EDIINT field group is added to the Trading Networks profiles. The EDIINT field group contains extended profile fields used for EDIINT processing. For more information about these fields, see [“Extended Fields Tab of the Profile”](#) on page 40.


Processing Rules

The EDIINT Module provides processing rules for EDIINT *transport-level* processing. If you want to do *business-level* processing on the payload of the EDIINT document, you need to:

- Configure EDIINT Module to submit the payload to Trading Networks. For instructions, see [“Configuring Whether Trading Networks Is to Process Payloads”](#) on page 53.
- Create your own processing rules to process the payloads. For information about creating processing rules, see *Building B2B Integrations: webMethods Trading Networks Administrator’s Guide*.

The following table describes the processing rules provided for EDIINT transport-level processing. For more information about the corresponding services for each processing rule, see [“Services Invoked by Processing Rules”](#) on page 60.

Processing Rule Name	Description
EDIINT Process Message	Trading Networks invokes this processing rule for inbound EDIINT documents. This processing rule invokes the <code>wm.EDIINT.rules:processMsg</code> service to process the inbound EDIINT document.
EDIINT Process MDN Message	Trading Networks invokes this processing rule for inbound EDIINT MDNs. This processing rule invokes the <code>wm.EDIINT.rules:processMDN</code> service to process an inbound MDN message.
EDIINT Send Message	Trading Networks invokes this processing rule for outbound EDIINT documents. This processing rule invokes the <code>wm.EDIINT.rules:sendMsg</code> service to initiate the sending of an outbound EDIINT document.
EDIINT Send MDN Message	Trading Networks invokes this processing rule for outbound EDIINT MDNs. This processing rule invokes the <code>wm.EDIINT.rules:sendMDN</code> service to initiate the sending of an outbound EDIINT MDN message.

 Important! You should *not* modify or customize these processing rules in any way.

Services Invoked by Processing Rules

The following table describes the services that are invoked by the processing rules described in [“Processing Rules”](#) above. All services are located in the `wm.EDIINT.rules` folder except for `deliveryDocument`, which is located in the `wm.EDIINT.delivery` folder.


Service	Description	Processing Rule that Invokes It
<code>processMsg</code>	Processes an inbound EDIINT document.	EDIINT Process Message
<code>processPayload</code>	Processes the payload of an EDIINT document. If you configure the EDIINT Module to process payloads, the <code>wm.EDIINT.rules:processMsg</code> service invokes this service to submit the payload to Trading Networks. For more information about configuring EDIINT Module to process payloads, see “Configuring Whether Trading Networks Is to Process Payloads” on page 53.	EDIINT Process Message
<code>processMDN</code>	Processes an inbound EDIINT MDN.	EDIINT Process MDN Message
<code>sendMsg</code>	Initiates the sending of an outbound EDIINT document.	EDIINT Send Message
<code>sendMDN</code>	Initiates the sending of an outbound EDIINT MDN document.	EDIINT Send MDN Message
<code>deliveryDocument</code>	Sends an outbound EDIINT document or MDN. The <code>wm.EDIINT.rules:sendMDN</code> and the <code>wm.EDIINT.rules:sendMsg</code> services invoke this service.	EDIINT Send Message -AND- EDIINT Send MDN Message

4 Creating a Client to Submit a Document Using EDIINT

■ Introduction	62
■ Content Types to Use	62
■ Setting the Input Variables for the <code>wm.EDIINT:send</code> Service	62

Introduction

The EDIINT standard requires the documents you send using the EDIINT transport be “packaged” in a specific way. If your client is running on the Integration Server that has the webMethods EDIINT Module installed, the client should invoke the `wm.EDIINT:send` service to package the document correctly and send it. If you are not using webMethods software for the client, see the documentation for the EDIINT software you are using to determine how to correctly package and send documents. This chapter describes how to create a client using the `wm.EDIINT:send` service of the EDIINT Module.

 Note: You can use the `wm.EDIINT:send` service to send both EDI documents and non-EDI documents.

Content Types to Use

You can use any of the EDIINT content types listed below. These content types are for both EDIINT documents and MDNs.

- multipart/signed
- multipart/report
- message/disposition-notification
- application/pkcs7-signature
- application/pkcs7-mime
- application/edi-X12
- application/EDIFACT
- application/XML
- application/edi-consent

Setting the Input Variables for the `wm.EDIINT:send` Service

The client should invoke the `wm.EDIINT:send` service to correctly package a document for EDIINT transport. For the list of input variables that the client should set for the `wm.EDIINT:send` service, see [Chapter 8, “webMethods EDIINT Module Services”](#).

5 Processing Inbound EDIINT Documents and MDNs

■ Processing Inbound EDIINT Documents	64
■ Processing Inbound EDIINT MDNs	67

more information, see [“How the S/MIME Type Profile Field Affects Processing Payloads” on page 66](#). For more information about setting up profiles for EDIINT, see [“Including EDIINT Information in Profiles” on page 36](#).

- 3 Set up Trading Networks to process the payload and perform the business-level logic.
 - If the payload is an EDI document, see the *webMethods EDI Module Installation and User's Guide* for how to set up the EDI Module to process the EDI documents.
 - If the payload is not an EDI document, you must:
 - Define a TN document type for the payload. For instructions, see the chapter about TN document types in *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.
 - Define a processing rule for the payload document. For instructions, see the chapter about processing rules in *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*.

How the S/MIME Type Profile Field Affects Processing Payloads

The EDIINT Module determines the S/MIME type used by the inbound EDIINT document, that is whether the inbound document is plain, signed, encrypted, or signed and encrypted. The EDIINT Module then compares the S/MIME type of the inbound document with the value of the S/MIME Type extended profile field of the sender's profile. The table below describes the actions the EDIINT Module takes based on the outcome of the comparison.

Value of S/MIME Type Extended Profile Field	S/MIME Type of Inbound Document	Action the EDIINT Module Takes for the Inbound Document
plain	any value	Processes the payload.
signed	signed signed and encrypted	Processes the payload.
	plain encrypted	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.
encrypted	encrypted signed and encrypted	Processes the payload.
	plain signed	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.
signedAndEncrypted	signed and encrypted	Processes the payload.
	plain signed encrypted	Logs an error message to the Trading Networks activity log and does <i>not</i> process the payload.

Processing Inbound EDIINT MDNs

The EDIINT Module provides the `wM.EDIINT:receive` service as the entry point for inbound EDIINT MDNs. That is, clients that send EDIINT MDNs must invoke the `wM.EDIINT:receive` service. For more about creating a client that sends EDIINT documents, see [Chapter 4, “Creating a Client to Submit a Document Using EDIINT”](#).

You do *not* need to create or customize services to process inbound EDIINT MDNs. The EDIINT Module provides all the logic needed to perform the processing.

Before You Can Process Inbound EDIINT MDNs

Define a profile for the sender and receiver of the EDIINT MDN. For instruction about how to create profiles, see the chapter about creating profiles in *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*. For information about adding EDIINT information to profiles, see [“Including EDIINT Information in Profiles”](#) on page 36.

Example of an EDIINT MDN Posted by HTTP

```
AS2-From: 987654321
AS2-To: 123456789
Message-ID: <2038921766.1012252564086.JavaMail.zhenzhou@zhenzhou>Content-Type:
multipart/signed; protocol="application/pkcs7-signature"; micalg=SHA-1;
boundary="-----_Part_20_-1967424986.1012252564076"
-----=_Part_20_-1967424986.1012252564076Content-Type: multipart/report; Report-
Type=disposition-notification; boundary="-----_Part_19_568293921.1012252564056"
-----=_Part_19_568293921.1012252564056Content-Type: text/plainContent-Transfer-
Encoding: 7bit
MDN for -
  Message ID: <128678451.1012252560430.JavaMail.zhenzhou@zhenzhou>
  From: 123456789
  To: 987654321
  Received on: 2002-01-28 at 16:16:04 (EST)
  Status: processed
  Comment: This is not a guarantee that the message has been completely processed or
understood by the receiving translator
-----=_Part_19_568293921.1012252564056Content-Type: message/disposition-
notificationContent-Transfer-Encoding: 7bit
Reporting-UA: webMethods Integration ServerOriginal-Recipient: 987654321Final-
Recipient: 987654321Original-Message-ID:
<128678451.1012252560430.JavaMail.zhenzhou@zhenzhou>Received-content-MIC:
qZvJD2+2H/OAQYa3+uIZUIyNUaw=, SHA-1Disposition: automatic-action/MDN-sent-
automatically; processed
-----=_Part_19_568293921.1012252564056--
-----=_Part_20_-1967424986.1012252564076Content-Type: application/pkcs7-signature;
name=smime.p7sContent-Transfer-Encoding: base64Content-Disposition: attachment;
filename=smime.p7s
MIAGCSqGSIB3DQEHAqCAMIACAQExCzAJBgUrDgMCGGUAMIAGCSqGSIB3DQEHAQAAMYIBujCCAbYCAQEwYDBb
MQswCQYDVQQGEwJVUzEXMBUGA1UEChMod2ViTWV0aG9kcyBJbmMxZDZANBgNVBASTB1BEIEVESTEiMCAGA1UE
AxMZRUJJSU5UIHNhbXBsZSBSZW1aXZ1ciBDQQIBATAJBgUrDgMCGGUAAoIGxMBGCGCSqGSIB3DQEFJAZELBgkq
hkiG9w0BBwEwHAYJKoZIhvcNAQkFMQ8XDTAyMDEyODIxMTYwNFowIwYJKoZIhvcNAQkEMRYEFPO/GE3KNoRk
F6KUtnqD0m40bUxEMFIFGCSqGSIB3DQEFJdZFFMEMwCgYIKoZIhvcNAwcwDgYIKoZIhvcNAwICAgCAMAOGCCqG
SIB3DQMCAgFAMAOGCCqGSIB3DQMCAgEoMAcGBS0AwIHMAOGCSqGSIB3DQEBAQUABIGAJBb3whwo+h0PsmEy
```

PMXQHIpjFS5fa5w8PI ipHQ9nfJVOTTbp5VTL4zT1E34vjESoktGBYmYnD+gTTe2aEB3PoIqCym25Lv2MZuv
SVNoa2hS4hrCnDwmYNqbFyS1V2ZAqodgBE1ztd71eeIgnXLU1/R65gF0Jw72Wto0xi8Q930AAAAAAA=
-----_Part_20_-1967424986.1012252564076--

6 Using EDIINT to Deliver Outbound Documents

■ Before You Can Deliver Outbound EDIINT Documents	70
■ Setting the S/MIME Type of the Outbound EDIINT Document	70
■ Using the wM.EDIINT:send Service to Send EDIINT Documents	70

Before You Can Deliver Outbound EDIINT Documents

Define a profile for the sender and receiver of the EDIINT document. For instruction about how to create profiles, see the chapter about creating profiles in *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*. For information about adding EDIINT information to profiles, see [“Including EDIINT Information in Profiles” on page 36](#).

Setting the S/MIME Type of the Outbound EDIINT Document

You can define the S/MIME type that you want the webMethods EDIINT Module (EDIINT Module) to use for an outbound EDIINT document; that is, whether you want to send the outbound EDIINT message:

- Without signing or encrypting (`plain`)
- Signing only (`signed`)
- Encrypting only (`encrypt`)
- Signing and encrypting (`signedAndEncrypted`)

You specify the S/MIME type that you want the EDIINT Module to use by using one of the following:

- The `type` input variable to the `wm.EDIINT:send` service
- The S/MIME Type extended profile field and setting the `type` input variable to the `wm.EDIINT:send` service to `getFromProfile`. For more information about setting the S/MIME Type extended profile field, see [“Extended Fields Tab of the Profile” on page 40 in Chapter 3, “Before You Can Transport Documents Using EDIINT”](#).


Using the `wm.EDIINT:send` Service to Send EDIINT Documents

The EDIINT Module provides the `wm.EDIINT:send` service to send EDIINT documents. This service performs all the necessary EDIINT transport-level processing.

To use the `wm.EDIINT:send` service:

- Set the `wm.EDIINT:send` service input variables as necessary. You must create a `java.io.InputStream` from the EDI or XML data and pass it the input variable `data/stream`. For a description of key input variables, see [“Setting the Input Variables for the `wm.EDIINT:send` Service” on page 62](#). For a complete description of this service, see [Chapter 8, “webMethods EDIINT Module Services”](#).

- Invoke the `wm.EDIINT:send` service from a service that you create.

 Important! Do *not* invoke the `wm.EDIINT:send` service directly from the Trading Networks Execute a Service processing action. Doing so will produce processing errors. The service that you create to invoke the `wm.EDIINT:send` service can be invoked directly from the Execute a Service processing action.

7 Viewing and Managing Information About EDIINT Documents and MDNs

■ Viewing Information about EDIINT Documents and MDNs	74
■ Resubmitting EDIINT Outbound Transactions	76

Viewing Information about EDIINT Documents and MDNs

Because EDIINT documents and MDNs are processed through Trading Networks, the EDIINT Module takes advantage of Trading Networks features.

One of the features is to use the Trading Networks system attribute, User Status, to assign a user-defined status to a document. The EDIINT Module assigns statuses to EDIINT documents and MDNs as it processes the document. For more information about viewing the user status associated with a document, see [“Viewing the User Status Associated with a Document”](#), below.

If you submit the payload of an inbound EDIINT document to Trading Networks for business-level processing, you can view information about the envelope(s) that the payload contains. For more information, see [“Viewing Related Documents”](#) on page 75.

Viewing the User Status Associated with a Document

You can view the User Status associated with a document on the Transactions page of My webMethods. For instructions on how to view information about documents, see *Managing B2B Integrations: webMethods Trading Networks User’s Guide*.

The following table describes the values of the User Status system attribute for EDIINT transport-level processing.

Type	User Status	Description
Inbound EDIINT document	ProcessMsg	The EDIINT document has been received and processing is starting.
	ProcessMsg PAYLOAD	The EDIINT document was successfully processed. The EDIINT Module is configured to process payloads and the <code>wm.EDIINT.rules:processPayload</code> service has been invoked.
	ProcessMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The EDIINT document contained an invalid sender ID or receiver ID. ■ The message could not be decrypted or verified.

Type	User Status	Description
Inbound EDIINT MDN	ProcessMDNMsg	The EDIINT MDN has been received and processing is starting.
	ProcessMDNMsg DONE	The MDN was processed.
	ProcessMDNMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The MDN contained an invalid sender ID or receiver ID. ■ The signature of the MDN could not be verified. ■ The MDN contained errors. ■ The MDN digest did not match that of the original EDIINT document.
	ProcessMDNMsg IGNORED	An identical MDN was previously received.
Outbound EDIINT document	SendMsg	Processing to send the EDIINT document has started.
	SendMsg DONE	One of the following: <ul style="list-style-type: none"> ■ The message was sent and an MDN was not requested. ■ The message was sent and an MDN was returned.
	SendMsg WAITMDN	The message was sent and an MDN was requested but not yet received.
	SendMsg ERROR	One of the following: <ul style="list-style-type: none"> ■ The message could not be sent. ■ The returned MDN contained errors.
Outbound EDIINT MDN	SendMDNMsg	Processing to send the EDIINT MDN has started.
	SendMDNMsg DONE	The MDN was sent successfully.
	SendMDNMsg ERROR	The MDN could not be sent.

Viewing Related Documents

If you submit the payload of an inbound EDIINT document to Trading Networks for business-level processing, you can view information about the envelope(s) that the payload contains.

By default Trading Networks persists to the Trading Networks database any EDIINT document that is submitted to it. If the document, which is stored in the pipeline variable *bizdoc*, has *not* been persisted (or if the `wm.EDIINT.rules:receive` service fails to place the document into *bizdoc*), the following occurs depending on the content type of the payload:

Payload Content Type	Action If the bizdoc Has Not Been Persisted
application/XML	No information about the payload is displayed.
application/edi-X12 application/EDIFACT application/edi-consent	Information about the main envelope is <i>not</i> displayed; information about each individual envelope is displayed if they have been persisted.
other	If you specified a user-defined service to handle another content type (as described in “Configuring Whether Trading Networks Is to Process Payloads” on page 53), the webMethods EDIINT Module passes the <i>bizdoc</i> to the user-defined service, along with the payload’s data stream and content type. The user-defined service must handle the display of payload information.

For instructions on how to view related documents in My webMethods, see *Managing B2B Integrations: webMethods Trading Networks User’s Guide*.

Resubmitting EDIINT Outbound Transactions

You can resubmit EDIINT outbound transactions. When you resubmit a document, Trading Networks performs the following processing:

- Uses the TN document type definitions to recognize the document
- Performs a processing rule lookup to determine the rule to use
- Performs the pre-processing and processing actions identified in the matching processing rule

Resubmit an outbound transaction if the document was not recognized when it was originally received. Create or modify a TN document type definition to recognize the document before processing the document again.

For instructions on how to resubmit documents in My webMethods, see *Managing B2B Integrations: webMethods Trading Networks User’s Guide*.

8 webMethods EDIINT Module Services

■ Overview	78
■ Summary of Elements in this Folder	78

Overview

Use the services in this folder for sending and receiving EDIINT (AS1, AS2, or AS3) messages and MDNs.

Summary of Elements in this Folder

The following elements are available in the public folder:

Element	Description
wm.EDIINT:receive	Receives inbound EDIINT (AS1/AS2/AS3) messages or MDNs and submits the message to Trading Networks to be unwrapped, decrypted, and signature authenticated.
wm.EDIINT:retrieveAS3Message	Downloads AS3 messages or MDNs from a partner's remote FTP server and submits the message to Trading Networks to be unwrapped, decrypted, and to have its signature authenticated.
wm.EDIINT:send	Constructs an outbound EDIINT message according to the configuration of the input variables, and then submits the message to Trading Networks.

wm.EDIINT:receive

Receives inbound EDIINT (AS1/AS2/AS3) messages or MDNs and submits the message to Trading Networks to be unwrapped, decrypted, and signature authenticated.

The EDIINT content handler populates the inputs described below.

Input Parameters

<i>protocol</i>	String The EDIINT protocol to use.								
	<table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>smtp</td><td>AS1 message or MDN.</td></tr><tr><td>http</td><td>AS2 message or MDN.</td></tr><tr><td>ftp</td><td>AS3 message or MDN.</td></tr></tbody></table>	Value	Meaning	smtp	AS1 message or MDN.	http	AS2 message or MDN.	ftp	AS3 message or MDN.
Value	Meaning								
smtp	AS1 message or MDN.								
http	AS2 message or MDN.								
ftp	AS3 message or MDN.								
<i>message-ID</i>	String The EDIINT message ID of the EDIINT message or MDN. This value becomes the Trading Networks system attribute, Document ID.								
<i>contentType</i>	String The content type of the EDIINT message or MDN.								

<i>stream</i>	Object The data InputStream representing the inbound EDIINT message or MDN.
<i>AS2-From</i>	String The sender ID from the EDIINT AS2 message/MDN. This should match the sender's Trading Networks external ID.
<i>AS2-To</i>	String The receiver ID from the EDIINT AS2 message/MDN. This should match the receiver's Trading Networks external ID.
<i>AS3-From</i>	String The sender ID from the EDIINT AS3 message/MDN. This should match the sender's Trading Networks external ID.
<i>AS3-To</i>	String The receiver ID from the EDIINT AS3 message/MDN. This should match the receiver's Trading Networks external ID.
<i>From</i>	String The sender ID from the EDIINT AS1 message/MDN. This should match the sender's Trading Networks external ID.
<i>To</i>	String The receiver ID from the EDIINT AS1 message/MDN. This should match the receiver's Trading Networks external ID.
<i>ReceiptDeliveryOption</i>	String The address to which to send an asynchronous MDN, if requested.

Output Parameters

None.

Usage Notes

For information about how to use this service, see [Chapter 5, "Processing Inbound EDIINT Documents and MDNs"](#).

wm.EDIINT:retrieveAS3Message

Downloads AS3 messages or MDNs from a partner's remote FTP server and submits the message to Trading Networks to be unwrapped, decrypted, and to have its signature authenticated.

The service locates the partner's remote FTP server using the values defined in IS document type `wm.EDIINT.TPA:EDIINTAS3TPA`. If the IS document type specifies that the retrieved file(s) are to be deleted, the service deletes the file(s).

Input Parameters

sender Document Optional. Identification of the partner from whom to retrieve the message.

Note: When *sender* is not specified, the service will access the remote FTP servers of *all* partners with whom you have a Trading Partner Agreement.

Variable	Meaning
----------	---------

<i>id</i>	String The partner's external ID.
-----------	-----------------------------------

<i>idTypeDesc</i>	String Optional. The partner's external ID type. Default: AS3.
-------------------	----------------------------------------------------------------

receiver Document Identification of the partner who retrieves the message.

Variable	Meaning
----------	---------

<i>id</i>	String The partner's external ID.
-----------	-----------------------------------

<i>idTypeDesc</i>	String Optional. The partner's external ID type. Default: AS3.
-------------------	----------------------------------------------------------------

Output Parameters

None.

wm.EDIINT:send

Constructs an outbound EDIINT message according to the configuration of the input variables, and then submits the message to Trading Networks.

Input Parameters

type String The S/MIME type that you want to use for the outbound EDIINT message. Specify one of the following:

Value	Description
-------	-------------

plain	Neither sign nor encrypt the outbound EDIINT message.
-------	-------------------------------------------------------

signed	Sign the outbound EDIINT message.
--------	-----------------------------------

encrypted	Encrypt the outbound EDIINT message.
-----------	--------------------------------------

	signedAndEncrypted	Sign and encrypt the outbound EDIINT message.
	getFromProfile	Determine whether to sign and/or encrypt the outbound EDIINT message using the value of the S/MIME Type extended field from the receiver's profile.
<i>compressed</i>	String (optional)	Whether you want the EDIINT message that you are sending to be compressed before it is signed and/or encrypted. Specify one of the following:
	<u>Value</u>	<u>Meaning</u>
	true	Compress the outbound message before signing and encrypting.
	false	Do <i>not</i> compress the outbound message before signing and encrypting. This is the default.
	getFromProfile	Determine whether to compress the outbound message using the value of the Compression extended field from the receiver's profile.
<i>deliveryMethod</i>	String	The delivery method you want to use to send the EDIINT document. The wm.EDIINT:send service obtains the Trading Networks profile for the receiver (specified by the <i>receiverID</i> input variable) and delivers the EDIINT document to the receiver's system using the information specified for the delivery method. Specify a delivery method that is defined in the receiver's Trading Networks profile.
	■	For EDIINT AS1, specify one of the following: <ul style="list-style-type: none"> ■ PrimarySMTP (corresponds to the Trading Networks Primary E-mail delivery method) ■ SecondarySMTP (corresponds to the Trading Networks Secondary E-mail delivery method)
	■	For EDIINT AS2, specify one of the following: <ul style="list-style-type: none"> ■ PrimaryHTTP ■ SecondaryHTTP ■ PrimaryHTTPS ■ SecondaryHTTPS

- For EDIINT AS3, specify AS3.

You can also specify `getFromProfile` if you want the service to obtain the delivery method from the Delivery Method extended field from the receiver's profile.

data

Document The payload that you want to send.

Variable	Description
<i>contentType</i>	The content type to assign to the outbound message.
<i>stream</i>	The <code>java.io.InputStream</code> that you map from the EDI or XML data.
<i>otherHeaders</i>	(optional) The name and value of the header for outbound messages. This header information is saved for each inbound payload document in Trading Networks in the <code>payloadMimeHeaders</code> content part.

requestMDN

String Whether you want the receiver to return an MDN. Specify one of the following:

Value	Meaning
<code>none</code>	Do <i>not</i> request a return MDN.
<code>synchronousMDN</code>	Request a return synchronous MDN.
<code>asynchronousMDN</code>	Request a return asynchronous MDN.
<code>getFromProfile</code>	Determine whether and what type of return MDN to request using the value of the Request MDN extended field from the receiver's profile.

Note: If you specify `PrimarySMTP`, `SecondarySMTP`, `PrimaryFTPS`, or `SecondaryFTPS` for *deliveryMethod*, you can only receive an asynchronous MDN.

requestSignedReceipt

String Whether you want the MDN to be signed.

Note: *requestSignedReceipt* is ignored when *requestMDN* is false.

Value	Meaning
<code>true</code>	Request a signed MDN.
<code>false</code>	Request a plain (unsigned) MDN.

	<code>getFromProfile</code>	Determine whether the MDN should be signed using the value of the Request Signed Receipt extended field from the receiver's profile.
<i>senderID</i>	Document Identification	of the sender of the EDIINT message.
	Variable	Description
	<i>id</i>	The sender's external ID. That is, the identification that you want for the sender in the message.
	<i>idTypeDesc</i>	Optional. The external ID type for the sender ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.
		Note: By default, the service uses the appropriate external ID type based on the value you specify for the <i>deliveryMethod</i> parameter. For example, if you specify <i>AS3</i> for the <i>deliveryMethod</i> parameter, the service uses the EDIINT <i>AS3</i> external ID type. Specify a value for <i>idTypeDesc</i> only if you want to override this default.
<i>receiverID</i>	Document Identification	of the receiver of the EDIINT message.
	Variable	Description
	<i>id</i>	The receiver's external ID. That is, the identification that you want for the receiver in the message.
	<i>idTypeDesc</i>	Optional. The external ID type for the receiver ID you specified in <i>id</i> . This is an external ID type as defined in Trading Networks.
		Note: By default, the service uses the appropriate external ID type based on the value you specify for the <i>deliveryMethod</i> parameter. For example, if you specify <i>AS3</i> for the <i>deliveryMethod</i> parameter, the service uses the EDIINT <i>AS3</i> external ID type. Specify a value for <i>idTypeDesc</i> only if you want to override this default.

ConversationID String (optional) Conversation ID for the outbound EDIINT message.

The *conversationID* parameter is an identifier that links all documents that are part of the same business process (also called a conversation). That is, all documents in the same business process need to have the same *conversationID*. As described in *Building B2B Integrations: webMethods Trading Networks Administrator's Guide*, Trading Networks can extract *conversationIDs* from EDI documents and use them to pass documents to the webMethods Process Engine after Trading Networks performs the actions identified by a processing rule.

In this field, you might want to specify the same conversation ID as that of the payload that you are sending. The EDIINT Module automatically assigns to a return MDN the same conversation ID that is assigned here.

Note: Leave this field blank unless you own a license for the webMethods Process Engine.

TNFlags Document Data to pass as input to the `wm.tn.route:routeBizDoc` service (for example, `TN_parms`).

<u>Variable</u>	<u>Description</u>
<i>priIgnoreDocument</i>	String Specify whether the <code>wm.tn.route:routeBizDoc</code> service should invoke the <code>pub.prt.tn:handleBizDoc</code> service (<code>false</code>) or not (<code>true</code>). Set this variable to <code>true</code> if the <code>WmPRT</code> package is not installed.

Output Parameters

None.

Usage Notes

For information about how to use this service, see [Chapter 4, "Creating a Client to Submit a Document Using EDIINT"](#) and [Chapter 6, "Using EDIINT to Deliver Outbound Documents"](#).

A Glossary

activity log

A log that webMethods Trading Networks (Trading Networks) maintains in its database to record activity that occurs within the Trading Networks system.

AS1

See [EDIINT AS1](#).

AS2

See [EDIINT AS2](#).

AS3

See [EDIINT AS3](#).

bizdoc

The name of the variable in the pipeline that contains the [BizDocEnvelope](#).

BizDocEnvelope

A BizDocEnvelope represents a routable Trading Networks transaction. It contains a document that Trading Networks is processing and includes additional information that Trading Networks requires for routing and processing the document. It is in the pipeline in the *bizdoc* variable and conforms to the IS document type `wm.tn.rec: BizDocEnvelope`.

business process

A multi-step interaction among participating systems, people, and trading partners. A business process can be fully automated (involve only interaction among computer systems) or include varying degrees of human interaction (for example, review and approval steps). It may be brief or long running. Some business processes transpire over days or weeks.

conversation

A specific case of a [business process](#) that involves a series of related documents being exchanged by two or more trading partners. All documents from a specific trading partner contain the same conversation ID. You model a conversation by creating a [process model](#) using webMethods Designer.

conversation ID

A webMethods Trading Networks' system document attribute that identifies a value within a document that is common to all documents that are part of the same [business process](#) (also called a [conversation](#)).

document type

See [IS Document Type](#) or [TN document type](#).

EDIINT

A standard defined by the Internet Engineering Task Force (IETF) that defines a protocol for using the Internet to securely exchange business documents (EDI, XML, or other). EDIINT stands for "Electronic Data Interchange-Internet Integration" or "EDI over the

Internet.” EDIINT has three version: *EDIINT AS1*, *EDIINT AS2*, and *EDIINT AS3*.

EDIINT AS1

EDIINT Applicability Statement 1, which is a version of the EDIINT standard that uses SMTP (e-mail) to transport documents.

EDIINT AS2

EDIINT Applicability Statement 2, which is a version of the EDIINT standard that uses HTTP (or HTTP/S) to transport documents.

EDIINT AS3

EDIINT Applicability Statement 3, which is a version of the EDIINT standard that uses FTP over SSL (also known as FTPS) to transport documents.

extended fields

Fields within a profile that are not provided with Trading Networks out of the box. Users of Trading Networks can define extended profile fields to extend the profile to meet their needs. Additionally, the EDI Module adds extended fields to profiles for use when transporting documents using EDIINT.

external ID

The value of an *external ID type*. For example, if an external ID type is a D-U-N-S number, the external ID is the actual D-U-N-S number.

external ID type

A type of identifier that trading partners use in documents. For example, a user might use a D-U-N-S number. The external ID type corresponds to the EDI ID qualifier in an EDI document.

IData object

The collection of name/value pairs on which a service operates. An IData object can contain any number of elements of any valid Java objects, including additional IData objects and IDataCodable objects.

IS Document Type

An element in the Integration Server’s namespace that contains a set of fields used to define the structure and type of data in an IS document (IData object).

MDNs

EDIINT receipts for messages that are received. The receiver sends the receipt back to the sender. MDN stands for message disposition notifications.

pipeline

The general term used to refer to the data structure in which input and output values are maintained for a flow service at run time. The pipeline starts with the input to the flow service and collects inputs and outputs from subsequent services in the flow. When a service in the flow executes, it has access to all data in the pipeline.

Process Engine

A facility of the Integration Server that manages the execution of processes (or *conversations*). You model a process (or conversation) using webMethods Designer.

processing rule

A webMethods Trading Networks object that contains set of actions that determine how Trading Networks is to process a document and criteria that indicates when to select a processing rule for an incoming document.

process model

Diagrams that illustrate and define the actions to perform for a *business process* or *conversation*. You create process models using webMethods Designer.

profile

A webMethods Trading Networks object that contains a summary of information about a corporation that is part of a trading network. A profile contains standard fields that Trading Networks defines and extended fields that are site defined.

TN document type

A webMethods Trading Networks object that defines how Trading Networks is to recognize a document and specifies initial actions to take on a recognized document.

